

N O T I C E

THIS DOCUMENT HAS BEEN REPRODUCED FROM
MICROFICHE. ALTHOUGH IT IS RECOGNIZED THAT
CERTAIN PORTIONS ARE ILLEGIBLE, IT IS BEING RELEASED
IN THE INTEREST OF MAKING AVAILABLE AS MUCH
INFORMATION AS POSSIBLE



National Aeronautics and
Space Administration

"Made available under NASA sponsorship
in the interest of early and wide dis-
semination of Earth Resources Survey
Program information and without liability
for any use made thereof."

JSC-13821
Revision A

Lyndon B. Johnson Space Center
Houston, Texas 77058

JAN 1980
80-10140
NASA CR-
160572

EARTH OBSERVATIONS DIVISION

SPACE AND LIFE SCIENCES DIRECTORATE

EARTH OBSERVATIONS DIVISION VERSION OF THE LABORATORY
FOR APPLICATIONS OF REMOTE SENSING SYSTEM
(EOD-LARSYS) USER GUIDE FOR THE
IBM 370/148
VOLUME IV - PROGRAM LISTINGS

Job Order 76-662

(E80-10140) EARTH OBSERVATIONS DIVISION
VERSION OF THE LABORATORY FOR APPLICATIONS
OF REMOTE SENSING SYSTEM (EOD-LARSYS) USER
GUIDE FOR THE IBM 370/148. VOLUME 4:
PROGRAM LISTINGS (Lockheed Electronics Co.) G3/43 N80-20727
Unclas 00140

Prepared By

Lockheed Electronics Company, Inc.
Systems and Services Division
Houston, Texas

Contract NAS 9-15800

November 1979



LEC-12566
Revision A

JSC-13821
Revision A

EARTH OBSERVATIONS DIVISION VERSION OF THE LABORATORY
FOR APPLICATIONS OF REMOTE SENSING SYSTEM
(EOD-LARSYS) USER GUIDE FOR THE
IBM 370/148
VOLUME IV - PROGRAM LISTINGS

Job Order 76-662

PREPARED BY


M. L. Burnell
Technical Publications Department

and


P. J. Aucoin
Earth Observations Data Products Department

APPROVED BY

NASA


J. M. Sulester, Technical
Monitor, Systems and
Facilities Branch

LEC


J. I. Morrow, Supervisor
Scientific Applications
Software Section

Prepared By

Lockheed Electronics Company, Inc.

For

Earth Observations Division

Space and Life Sciences Directorate

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
LYNDON B. JOHNSON SPACE CENTER
HOUSTON, TEXAS

November 1979

LEC-12566
Revision A

CONTENTS

Section	Page
1. SCOPE.	1-1
2. APPLICABLE DOCUMENTS	2-1
3. MONITOR	3-1
4. MSCAN	4-1
5. COMMON BLOCKS AND BLOCK DATA	5-1
6. HIST PROCESSOR	6-1
7. GRAYMAP PROCESSOR.	7-1
8. STAT PROCESSOR	8-1
9. ISOCLS PROCESSOR	9-1
10. SELECT PROCESSOR	10-1
11. CLASSIFY PROCESSOR	11-1
12. DISPLAY PROCESSOR.	12-1
13. DATA-TR PROCESSOR.	13-1
14. TRSTAT PROCESSOR	14-1
15. NDHIST PROCESSOR	15-1
16. SCTRPL PROCESSOR	16-1
17. DOTDATA PROCESSOR.	17-1
18. LABEL PROCESSOR	18-1
19. UTILITY SUBPROGRAMS.	19-1
20. DAMRG PROCESSOR	20-1
21. GTDDM PROCESSOR	21-1
22. GTTCN PROCESSOR	22-1
23. TESTSP PROCESSOR	23-1

1. SCOPE

This document is one of a four-volume series entitled "Earth Observations Division Version of the Laboratory for Applications of Remote Sensing System (EOD-LARSYS) User Guide for the IBM 370/148" (section 2). Originally, the EOD-LARSYS software was written for execution on the Univac 1108/1110 computer at the Laboratory for Applications of Remote Sensing (LARS). The original version of this document covers the conversion of the EOD-LARSYS software for execution on the IBM 370/148, which was acquired subsequently by the LARS. The LARS recently replaced the IBM 370/148 with the IBM 3031 computer, which is thoroughly compatible with software as altered for execution on the IBM 370/148. Thus, no conversion of software is required for this system to be operable on the IBM 3031.

This volume IV contains a listing for each subprogram within the existing EOD-LARSYS processors and the utility subroutines. It is modeled after the As-Built Documentation (volume III), inasmuch as the listings appear in the same order as the subprograms are documented in volume III. Table 1-1 of volume III lists the EOD-LARSYS subprograms in alphabetical order, along with the processor to which each belongs. The processors, by section, are as follows:

<u>Section</u>	<u>Processor</u>
6	One-Dimensional Histogram (HIST)
7	GRAYMAP
8	Statistics (STAT)
9	Iterative Self-Organizing Clustering System (ISOCLS)
10	Feature Selection (SELECT)
11	Classification (CLASSIFY)
12	Performance Display (DISPLAY)

<u>Section</u>	<u>Processor</u>
13	Data-Transformation (DATA-TR)
14	Statistics Transformation (TRSTAT)
15	N-Dimensional Histogram (NDHIST)
16	Scatter Plot (SCTRPL)
17	Dot Data (DOTDATA)
18	Automatic Cluster Labeling (LABEL)

Within each of the above sections, the processor driver routine is listed first, followed by the subprogram listings in alphabetical order (the same order as they are documented in revision A of volume III). Utility subprograms are listed in section 19. In addition, this document contains subprogram listings for the following new processors:

<u>Section</u>	<u>Processor</u>
20	Data Merge (DAMRG)
21	Ground Truth Data Tape Dump (GTDDM)
22	Ground Truth Tape Conversion (GTTCN)
23	Iterative Self-Organizing Clustering System Using Packed Pixel Storage (TESTSP)

The listing for the EOD-LARSYS monitor routine, MONTOR, is given in section 3, along with a listing for an optional monitor routine, MONPAC. Provisions have been made in MONTOR for the addition of the following processors to the system: CLASY, AMOEBA, Equi-Probable Blocks (EQUPRB), Multitemporal Bayes (MULBAY), and Principal Component Greenness (PCG). These, which will be a part of the EOD-LARSYS, will be documented separately.

The MONPAC routine was created for use with the TESTSP processor, which clusters pixel values and stores them in packed form on disk storage. It differs from MONTOR in that it stores pixels in packed form (one sample per byte) rather than in floating point (one sample value every four bytes), as is done by the ISOCLS processor. The MONPAC routine may be used with other processors.

The listing for MSCAN, MONTOR's supervisory routine, is given in section 4, and common block listings are given in section 5.

2. APPLICABLE DOCUMENTS

1. Stewart, J.; et al.: EOD-LARSYS User Guide for the IBM 370/148 - vol. I, System Overview. JSC-13821, LEC-12563, NASA/JSC (Houston), Aug. 1978.
2. Stewart, J.; et al.: EOD-LARSYS User Guide for the IBM 370/148 - vol. II, User's Reference Manual. JSC-13821, LEC-12564, NASA/JSC (Houston), Dec. 1978.
3. Burnell, M. L.; et al.: EOD-LARSYS User Guide for the IBM 370/148 - vol. III, As-Built Documentation. JSC-13821, LEC-12565, NASA/JSC (Houston), Mar. 1979. (Revision A to be published.)

3. MONITOR

FILE MONITOR

```

C      //MONITOR
C-----
C      CALL..   SYSTEM MONITOR (// EXEC LARSYSA )
C      PURPOSE.. MONITORS THE VARIOUS SYSTEM SUPERVISORS
C      ROUTINES  MSCAN  CLSFY  DSPLY  STAT
C                  SELECT HIST  GRAYMP  DATATR
C                  ISOCLS TRSTAT NDHIST SCTRPL
C                  DOTDATA LABEL  EQUFRB  MULRAY
C                  GTTCN  DAMRG
C                  AMOEBA CLASY TESTSP GTDDM PCG EXIT
C      RETURNS.. NONE
C-----
C      IMPLICIT INTEGER(A-H,O-Z)
C      COMMON ARRAY
C      DIMENSION ARRAY(10600)
C
C      *ARRAY* IS A BLOCK OF STORAGE PASSED TO EACH PROCESSOR FOR THE
C      VARIABLE DIMENSIONING OF OTHER ARRAYS. THE ARRAY IS NEVER USED
C      TO PASS INFORMATION FROM ONE PROCESSOR TO ANOTHER.
C
C      DATA TOP/10600/
C      INCLUDE COMRKG.LIST
C
C      INCLUDE COMNT6.LIST
C      COMMON/GLOBAL/HEAD(63),MAPTAP,DATAP,SAVTAP,BMFILE,BMKEY,
C      * HISFIL,HISKEY,TRFORM,ERIPTP,ERPKEY,MAPUNT,NOFILE,
C      * DRUMAD,DRMWD,DRMWS,PAGSIZ,DATFIL,STAFIL,ASAV,ASAVL
C      * ,NHSTUN,NHSTFI,SCTRUN,MAPFIL
C      * ,DOTUNT,DOTFIL,NCHPAS,TRNSFL,BMTRFL,HISTFL,PCHUNT,
C      * CRDUNT,PPTUNT,WANDIO
C
C      GLOBAL COMMON IS USED IN EVERY PROCESSOR. IT IS ALWAYS IN COPE.
C      ALL PARAMETERS ARE INITIALIZED IN THE MONITOR.ROUTINE OR BLKCOM
C      EXCEPT AS NOTED BELOW
C      DEFINITIONS
C      HEAD - STANDARD HEADING PRINTED ON MOST OUTPUT PAGES.
C      MAPTAP - FORTRAN UNIT NUMBER ON WHICH THE MAPTAP FILE IS
C              WRITTEN (=2)
C      DATAP - UNIT NO. FOR THE IMAGE DATA TAPE (=3)
C      SAVTAP - UNIT NO. ON WHICH THE STATISTICS FILE IS WRITTEN (=1)
C      BMFILE - UNIT NO. ON WHICH THE B-MATRIX FILE IS WRITTEN (=10)
C      BMKEY - TRIGGER INDICATING THAT THE B-MATRIX FILE HAS BEEN
C              WRITTEN. CAN BE SET IN SELECT CLASSIFY OR DATA-TR.
C      HISFIL - UNIT NO. ON WHICH THE HISTOGRAM FILE IS WRITTEN (=13)
C      HISKEY - TRIGGER INDICATING THE HISTOGRAM FILE HAS BEEN
C              WRITTEN. SET IN HIST PROCESSOR.
C      TRFORM - UNIT NO. ON WHICH THE TRANSFORMED IMAGE IS WRITTEN BY
C              THE DATA-TRANSFORMATION PROCESSOR. (=14)
C      ERIPTP - UNIT NO. ON WHICH THE ISOCLS PROCESSOR WRITES
C              CLUSTER STATISTICS FOR THE ERIPS SYSTEM. (=15)
C      ERPKEY - TRIGGER INDICATING THAT THE ERIPS INTERFACE TAPE
C              HAS BEEN WRITTEN.
C      MAPUNT - UNIT NO. ON WHICH THE ISOCLS OR DISPLAY PROCESSOR
C              WRITES THE CLUSTERED OR CLASSIFIED DATA
C              TO BE DISPLAYED ON THE PMIS DAS
C      NOFILE - NO. OF FILES WRITTEN ON UNIT 16 (MAP OUTPUT TAPE)
C              BY DISPLAY AND/OR ISOCLS
C              SET EITHER IN ISOCLS OR DISPLAY.
C      DRUMAD - BEGINNING ADDRESS FOR THE RANDOM ACCESS HIGH SPEED
C      REAL TIME
C              DRUM FILE. THIS FILE IS USED AS A SWATCH FILE IN
C              SEVERAL PROCESSORS. REFERENCES TO SYSTEM ROUTINES
C              'READ' AND 'WRITE' ACCESS THIS FILE.
C      DRMWD - NO. OF WORDS AVAILABLE ON THE RANDOM ACCESS FILE.
C      PAGSIZ - NO. OF LINES AVAILABLE FOR PRINTING ON A PAGE.
C      DATFIL - NO. OF E-O-F'S TO BE READ OVER BY TAPERD ROUTINE IN
C              ORDER TO POSITION THE DATA TAPE TO DESIRED FILE
C      STAFIL - NO. OF E-O-F'S TO SKIP OVER TO POSITION STAT FILE*)

```

MON00010
 MON00020
 MON00030
 MON00040
 MON00050
 MON00060
 MON00070
 MON00080
 MON00090
 MON00100
 MON00110
 MON00120
 MON00130
 MON00140
 MON00150
 MON00160
 MON00170
 MON00180
 MON00190
 MON00200
 MON00210
 MON00220
 MON00230
 MON00240
 MON00250
 MON00260
 MON00270
 MON00280
 MON00290
 MON00300
 MON00310
 MON00320
 MON00330
 MON00340
 MON00350
 MON00360
 MON00370
 MON00380
 MON00390
 MON00400
 MON00410
 MON00420
 MON00430
 MON00440
 MON00450
 MON00460
 MON00470
 MON00480
 MON00490
 MON00500
 MON00510
 MON00520
 MON00530
 MON00540
 MON00550
 MON00560
 MON00570
 MON00580
 MON00590
 MON00600
 MON00610
 MON00620
 MON00630
 MON00640
 MON00650
 MON00660
 MON00670
 MON00680
 MON00690
 MON00700
 MON00710
 MON00720
 MON00730
 MON00740
 MON00750
 MON00760

FILE MONITOR

```

C*      ASAV - UNIT NO. ON WHICH TRSTAT WRITES THE TRANSFORMED MON00770
C*      STATS MON00780
C*      ASAVFL - NO. OF E-O-F'S TO SKIP OVER TO POSITION TRANSFORMED MON00790
C*      STATS MON00800
C*      DOTUNT - UNIT NO. ON WHICH DOT DATA FILE (DOTFIL) IS WRITTEN MON00810
C*      DOTFIL - NO. OF E-O-F'S TO SKIP OVER TO POSITION DOTFIL FILE MON00820
C*      NCHPAS - NO. OF CHANNELS PER PASS MON00830
C*      TRNSFL - NO. OF E-O-F'S TO SKIP OVER FOR TRFORM FILE MON00840
C*      HMTXFL - NO. OF E-O-F'S TO SKIP OVER FOR HMFIL FILE MON00850
C*      HISTFL - NO. OF E-O-F'S TO SKIP OVER FOR HISFIL FILE MON00860
C*      PUNCH - UNIT NO. FOR CARD PUNCH FILE MON00870
C*      CRDUNT - UNIT NO. FOR CARD READER MON00880
C*      RANDIO - SCRATCH UNIT FOR RREAD AND RWRITE ROUTINES MON00890
CSEND MON00900
      DEBUG=-1 MON00910
C*      SYSTEM ROUTINE RINIT ASSIGNS THE RANDOM ACCESS DRUM FILE. MON00920
C*      -DRUMAD-- IS THE ADDRESS TO BEGIN WRITING MON00930
C*      -DRUMWDS- IS THE NO. OF WORDS AVAILABLE ON THE DRUM FILE. MON00940
C*      THE FOLLOWING PROCESSORS USE THE RANDOM ACCESS DRUM FILE FOR SCRATCH MON00950
C*      -ISOCLS- MON00960
C*      -DISPLY- MON00970
C*      -SELECT- MON00980
C*      -GRAYMP- MON00990
C*      -SIGEXT- MON01000
C*      DEFINE FILE 22(2100,200,U,1D) MON01010
C*      DRUMAD=1 MON01020
C*      DRUMWDS=420000 MON01030
C*      WRITE(22,1)DRUMAD MON01040
10  CONTINUE MON01050
C*      TIME = 0. MON01060
C*      CALL CLOCK(0) MON01070
C*      CALL MSCAN(JGO,DRUG) MON01080
C*      GO TO (20,40,60,80,100,120,140,160,175,180,200,220,240,260, MON01090
C*      * 280,290,300,310,320,330,340,350,360,370),JGO MON01100
C*      20 CONTINUE MON01110
C*      CALL STAT(ARRAY, TOP) MON01120
C*      CALL CLOCK ( 1, 'SSTA' ) MON01130
C*      GO TO 10 MON01140
C*      40 CONTINUE MON01150
C*      CALL CLSFY(ARRAY, TOP) MON01160
C*      CALL CLOCK ( 1, 'SCLA' ) MON01170
C*      GO TO 10 MON01180
C*      60 CONTINUE MON01190
C*      CALL DSPLY(ARRAY, TOP) MON01200
C*      CALL CLOCK ( 1, 'SDIS' ) MON01210
C*      GO TO 10 MON01220
C*      80 CONTINUE MON01230
C*      CALL SELECT(ARRAY, TOP) MON01240
C*      CALL CLOCK ( 1, 'SSEL' ) MON01250
C*      GO TO 10 MON01260
C*      100 CONTINUE MON01270
C*      CALL HIST(ARRAY, TOP) MON01280
C*      CALL CLOCK ( 1, 'SHIS' ) MON01290
C*      GO TO 10 MON01300
C*      GO HERE FOR ISOCLS MON01310
C*      120 CONTINUE MON01320
C*      CALL ISOCLS(ARRAY, TOP) MON01330
C*      CALL CLOCK ( 1, 'SISO' ) MON01340
C*      GO TO 10 MON01350
C*      130 MON01360
C*      MON01370
C*      MON01380
C*      MON01390
C*      MON01400
C*      MON01410
C*      MON01420
C*      MON01430
C*      MON01440
C*      MON01450
C*      MON01460
C*      MON01470
C*      MON01480
C*      MON01490
C*      MON01500
C*      MON01510
C*      MON01520

```

FILE MONITOR

C	GO HERE FOR GRAYMAP	MONO	530
140	CONTINUE	MONO	540
	CALL GRAYMP(ARRAY.TOP)	MONO	550
150	CALL CLOCK (1, 'SGRA')	MONO	560
	GO TO 10	MONO	570
C	GO HERE FOR DATA-TRANSFORMATION	MONO	580
160	CONTINUE	MONO	590
	CALL DATATR(ARRAY.TOP)	MONO	600
170	CALL CLOCK (1, 'SDAT')	MONO	610
	GO TO 10	MONO	620
C	GO HERE FOR SIGEXT MODULE	MONO	630
175	CONTINUE	MONO	640
***	SIGEXT	MONO	650
176	CALL CLOCK (1, 'SSIG')	MONO	660
	GO TO 10	MONO	670
C	GO HERE FOR THSTAT	MONO	680
180	CONTINUE	MONO	690
	CALL TRSTAT(ARRAY.TOP)	MONO	700
190	CALL CLOCK (1, 'STRS')	MONO	710
	GO TO 10	MONO	720
C	GO HERE FOR NDHIST	MONO	730
200	CONTINUE	MONO	740
	CALL NDHIST(ARRAY.TOP)	MONO	750
210	CALL CLOCK (1, 'SNDH')	MONO	760
	GO TO 10	MONO	770
C	GO HERE FOR SCTRPL	MONO	780
220	CONTINUE	MONO	790
	CALL SCTRPL(ARRAY.TOP)	MONO	800
230	CALL CLOCK (1, 'SCT')	MONO	810
	GO TO 10	MONO	820
C	GO HERE FOR DOTDATA	MONO	830
240	CONTINUE	MONO	840
	CALL DOTDAT(ARRAY.TOP)	MONO	850
250	CALL CLOCK (1, 'SDOT')	MONO	860
	GO TO 10	MONO	870
C	GO HERE FOR LABEL	MONO	880
260	CONTINUE	MONO	890
	CALL LABEL(ARRAY.TOP)	MONO	900
270	CALL CLOCK (1, 'SLAB')	MONO	910
	GO TO 10	MONO	920
C	GO HERE FOR EQUI-PROBABLE BLOCKS CLASSIFIER	MONO	930
280	CONTINUE	MONO	940
	CALL EQUIPR(ARRAY.TOP)	MONO	950
285	CALL CLOCK (1, 'SEQU')	MONO	960
	GO TO 10	MONO	970
C	GO HERE FOR MULTI-TEMPORAL CLASSIFIER	MONO	980
290	CONTINUE	MONO	990
	CALL MULHAY(ARRAY.TOP)	MONO	2000
295	CALL CLOCK (1, 'SMUL')	MONO	2010
	GO TO 10	MONO	2020
C	GO HERE FOR GROUND TRUTH TO MAPFIL	MONO	2030
300	CONTINUE	MONO	2040
	CALL GTTCN(ARRAY.TOP)	MONO	2050
305	CALL CLOCK (1, 'SGTT')	MONO	2060
	GO TO 10	MONO	2070
		MONO	2080
		MONO	2090
		MONO	2100
		MONO	2110
		MONO	2120
		MONO	2130
		MONO	2140
		MONO	2150
		MONO	2160
		MONO	2170
		MONO	2180
		MONO	2190
		MONO	2200
		MONO	2210
		MONO	2220
		MONO	2230
		MONO	2240
		MONO	2250
		MONO	2260
		MONO	2270
		MONO	2280

FILE MONTOR

```

C
C 310 GO HERE FOTR IMAGE DATA MENGE
      CONTINUE
      CALL DAMRG(ARRAY, TOP)
315   CALL CLOCK ( 1, 'SDAM' )
      GO TO 10
C
C 320 GO HERE FOR AMOEBA
      CONTINUE
      CALL AMOERA(ARRAY, TOP)
325   CALL CLOCK ( 1, 'SAMO' )
      GO TO 10
C
C 330 GO HERE FOR CLASY
      CONTINUE
      CALL CLASY(ARRAY, TOP)
335   CALL CLOCK ( 1, 'SCLS' )
      GO TO 10
C
C 340 GO HERE FOR TESTSP
      CONTINUE
      CALL TESTSP(ARRAY, TOP)
345   CALL CLOCK ( 1, 'STES' )
      GO TO 10
C
C 350 GO HERE FOR GROUND TRUTH DOT UNLOAD
      CONTINUE
      CALL GTDDM(ARRAY, TOP)
355   CALL CLOCK ( 1, 'SGTD' )
      GO TO 10
C
C 360 GO HERE FOR PCG
      CONTINUE
      CALL PGSTAT(ARRAY, TOP)
      CALL CLOCK (1, 'SPCG')
      GO TO 10
C
C 370 GO HERE TO EXIT
      IF (NOFILE .GT. 0) REWIND MAPUNT
      END

```

```

MON02290
MON02300
MON02310
MON02320
MON02330
MON02340
MON02350
MON02360
MON02370
MON02380
MON02390
MON02400
MON02410
MON02420
MON02430
MON02440
MON02450
MON02460
MON02470
MON02480
MON02490
MON02500
MON02510
MON02520
MON02530
MON02540
MON02550
MON02560
MON02570
MON02580
MON02590
MON02600
MON02610
MON02620
MON02630
MON02640
MON02650
MON02660
MON02670
MON02680
MON02690
MON02700
MON02710
MON02720
MON02730
MON02740

```


FILE: CLOCK FORTRAN A

```
SUBROUTINE CLOCK(TIME)
  INTEGER TIMER
  TIME1 = TIMER(0)
  TIME = (TIME1/40000.)-TIME
  RETURN
END
```

```
CL000010
CL000020
CL000030
CL000040
CL000050
CL000060
```

FILE MONPAC

//MONITOR

CALL.. SYSTEM MONITOR (// EXEC LARYSAA)
 PURPOSE.. MONITORS THE VARIOUS SYSTEM SUPERVISORS
 ROUTINES MSCAN CLSFY DPLAY STAT
 SELECT HIST GRAYMP DATATR
 ISUCLS TSTAT NUMIST SCTRPL
 DOTDATA LABEL EQUIPM MULRAY
 GTTCN DAMNG
 AMOEBA CLASY TESTSP GTDDM PCG EXIT
 RETURNS.. NONE

IMPLICIT INTEGER(A-M,O-Z)
 COMMON ARRAY
 DIMENSION ARRAY(10600)

ARRAY IS A BLOCK OF STORAGE PASSED TO EACH PROCESSOR FOR THE
 VARIABLE DIMENSIONING OF OTHER ARRAYS. THE ARRAY IS NEVER USED
 TO PASS INFORMATION FROM ONE PROCESSOR TO ANOTHER.

DATA TOP/10600/
 INCLUDE COMMK6.LIST

INCLUDE COMM16.LIST
 COMMON/GLOBAL/HEAD(163),MAPTAP,DATAP,SAVTAP,BMFILE,BMKEY,
 HISFIL,HISKEY,TFORM,ERITP,ERPKEY,MAPUNT,NOFILE,
 DRUMAD,DAMNGS,PAGSIZ,DATFIL,STAFIL,ASAV,ASAVFL
 HISTUNT,HISTFL,SCTRUN,MAPFIL
 DOTUNT,DOTFIL,NCHPAS,TRNSFL,BMTRFL,HISTFL,PCHUNT,
 CRDUNT,PRINTNT,MANDIO

GLOBAL COMMON IS USED IN EVERY PROCESSOR. IT IS ALWAYS IN CORE.
 ALL PARAMETERS ARE INITIALIZED IN THE MONITOR ROUTINE OR HLKCOM
 EXCEPT AS NOTED BELOW

DEFINITIONS

HEAD - STANDARD HEADING PRINTED ON MOST OUTPUT PAGES.
 MAPTAP - FORTRAN UNIT NUMBER ON WHICH THE MAPTAP FILE IS
 WRITTEN (=2)
 DATAP - UNIT NO. FOR THE IMAGE DATA TAPE (=3)
 SAVTAP - UNIT NO. ON WHICH THE STATISTICS FILE IS WRITTEN (=1)
 BMFILE - UNIT NO. ON WHICH THE H-MATRIX FILE IS WRITTEN (=10)
 BMKEY - TRIGGER INDICATING THAT THE H-MATRIX FILE HAS BEEN
 WRITTEN. CAN BE SET IN SELECT CLASSIFY OR DATA-TR.
 HISTFIL - UNIT NO. ON WHICH THE HISTOGRAM FILE IS WRITTEN (=13)
 HISKEY - TRIGGER INDICATING THE HISTOGRAM FILE HAS BEEN
 WRITTEN. SET IN HIST PROCESSOR.
 TFORM - UNIT NO. ON WHICH THE TRANSFORMED IMAGE IS WRITTEN BY
 THE DATA-TRANSFORMATION PROCESSOR. (=14)
 ERITP - UNIT NO. ON WHICH THE ISUCLS PROCESSOR WRITES
 CLUSTER STATISTICS FOR THE ERIPS SYSTEM. (=15)
 ERPKEY - TRIGGER INDICATING THAT THE ERIPS INTERFACE TAPE
 HAS BEEN WRITTEN.
 MAPUNT - UNIT NO. ON WHICH THE ISUCLS OR DISPLAY PROCESSOR
 WRITES THE CLUSTERED OR CLASSIFIED DATA
 TO BE DISPLAYED ON THE BMIS DAS
 NOFILE - NO. OF FILES WRITTEN ON UNIT 16 (MAP OUTPUT TAPE)
 BY DISPLAY AND/OR ISUCLS
 SET EITHER IN ISUCLS OR DISPLAY.
 DRUMAD - BEGINNING ADDRESS FOR THE RANDOM ACCESS HIGH SPEED
 REAL TIME
 DATA FILE. THIS FILE IS USED AS A SCRATCH FILE IN
 SEVERAL PROCESSORS. REFERENCES TO SYSTEM ROUTINES
 THEREAFTER AND 'WRITE' ACCESS THIS FILE.
 DAMNGS - NO. OF WORDS AVAILABLE ON THE RANDOM ACCESS FILE.
 PAGSIZ - NO. OF LINES AVAILABLE FOR PRINTING ON A PAGE.
 DATFIL - NO. OF E-O-F'S TO BE READ OVER BY TAPERO ROUTINE IN
 ORDER TO POSITION THE DATA TAPE TO DESIRED FILE
 STAFIL - NO. OF E-O-F'S TO SKIP OVER TO POSITION STAT FILE*)

MON00010
 MON00020
 MON00030
 MON00040
 MON00050
 MON00060
 MON00070
 MON00080
 MON00090
 MON00100
 MON00110
 MON00120
 MON00130
 MON00140
 MON00150
 MON00160
 MON00170
 MON00180
 MON00190
 MON00200
 MON00210
 MON00220
 MON00230
 MON00240
 MON00250
 MON00260
 MON00270
 MON00280
 MON00290
 MON00300
 MON00310
 MON00320
 MON00330
 MON00340
 MON00350
 MON00360
 MON00370
 MON00380
 MON00390
 MON00400
 MON00410
 MON00420
 MON00430
 MON00440
 MON00450
 MON00460
 MON00470
 MON00480
 MON00490
 MON00500
 MON00510
 MON00520
 MON00530
 MON00540
 MON00550
 MON00560
 MON00570
 MON00580
 MON00590
 MON00600
 MON00610
 MON00620
 MON00630
 MON00640
 MON00650
 MON00660
 MON00670
 MON00680
 MON00690
 MON00700
 MON00710
 MON00720
 MON00730
 MON00740
 MON00750
 MON00760

FILE MONPAC

```

C*      ASAV - UNIT NO. ON WHICH TRSTAT WRITES THE TRANSFORMED MON00770
C*      STATS MON00780
C*      ASAVFL - NO. OF E-O-F'S TO SKIP OVER TO POSITION TRANSFORMED MON00790
C*      STATS MON00800
C*      DOTUNT - UNIT NO. ON WHICH DOT DATA FILE (DOTFIL) IS WRITTEN MON00810
C*      DOTFIL - NO. OF E-O-F'S TO SKIP OVER TO POSITION DOTFIL FILE MON00820
C*      NCHPAS - NO. OF CHANNELS PER PASS MON00830
C*      TRANSFL - NO. OF E-O-F'S TO SKIP OVER FOR TRFORM FILE MON00840
C*      HMTXFL - NO. OF E-O-F'S TO SKIP OVER FOR BMFIL FILE MON00850
C*      HISTFL - NO. OF E-O-F'S TO SKIP OVER FOR HISFIL FILE MON00860
C*      PUNCH - UNIT NO. FOR CARD PUNCH FILE MON00870
C*      CPDUNT - UNIT NO. FOR CARD READER MON00880
C*      RANDIO - SCRATCH UNIT FOR RREAD AND RWRITE ROUTINES MON00890
C*      C$END MON00900
C*      DRUG=-1 MON00910
C*      SYSTEM ROUTINE RINIT ASSIGNS THE RANDOM ACCESS DRUM FILE. MON00920
C*      -DRUMAD-- IS THE ADDRESS TO BEGIN WRITING MON00930
C*      -DRMWDS- IS THE NO. OF WORDS AVAILABLE ON THE DRUM FILE. MON00940
C*      THE FOLLOWING PROCESSORS USE THE RANDOM ACCESS DRUM FILE FOR SCRAT MON00950
C*      -ISOCLS- MON00960
C*      -DISPLY- MON00970
C*      -SELECT- MON00980
C*      -GRAYMP- MON00990
C*      -SIGEXT- MON01000
C*      MON01010
C*      MON01020
C*      MON01030
C*      MON01040
C*      MON01050
C*      MON01060
C*      MON01070
C*      MON01080
C*      MON01090
C*      MON01100
C*      MON01110
C*      MON01120
C*      MON01130
C*      MON01140
C*      MON01150
C*      MON01160
C*      MON01170
C*      MON01180
C*      MON01190
C*      MON01200
C*      MON01210
C*      MON01220
C*      MON01230
C*      MON01240
C*      MON01250
C*      MON01260
C*      MON01270
C*      MON01280
C*      MON01290
C*      MON01300
C*      MON01310
C*      MON01320
C*      MON01330
C*      MON01340
C*      MON01350
C*      MON01360
C*      MON01370
C*      MON01380
C*      MON01390
C*      MON01400
C*      MON01410
C*      MON01420
C*      MON01430
C*      MON01440
C*      MON01450
C*      MON01460
C*      MON01470
C*      MON01480
C*      MON01490
C*      MON01500
C*      MON01510
C*      MON01520

      DEFINE FILE 22(640,200,U,10)
      DRUMAD=1
      DRMWDS=128000
      WRITE(22,1)DRUMAD
10  CONTINUE
      TIME = 0.
      CALL CLOCK(TIME)
      CALL MSCAN(J60,DRUG)
      GO TO (20,40,60,80,100,120,140,160,175,180,200,220,240,260,
      * 280,290,300,310,320,330,340,350,360,370),J60
C
20  CONTINUE
      CALL STAT(ARRAY, TOP)
30  CALL CLOCK ( 1, '$STA' )
      GO TO 10
C
40  CONTINUE
      CALL CLSEY(ARRAY, TOP)
50  CALL CLOCK ( 1, '$CLA' )
      GO TO 10
C
60  CONTINUE
      CALL DSPLY(ARRAY, TOP)
70  CALL CLOCK ( 1, '$DIS' )
      GO TO 10
C
80  CONTINUE
      CALL SELECT(ARRAY, TOP)
90  CALL CLOCK ( 1, '$SEL' )
      GO TO 10
C
100 CONTINUE
      CALL HIST(ARRAY, TOP)
110 CALL CLOCK ( 1, '$HIS' )
      GO TO 10
C
      GO HERE FOR ISOCLS
C
120 CONTINUE
      CALL ISOCLS(ARRAY, TOP)
130 CALL CLOCK ( 1, '$ISO' )
      GO TO 10
C

```

FILE MONPAC

```

C      GO HERE FOR GRAYMAP
140    CONTINUE
      CALL GRAYMP (ARRAY, TOP)
150    CALL CLOCK ( 1, '$GRA' )
      GO TO 10
C
C      GO HERE FOR DATA-TRANSFORMATION
C
160    CONTINUE
      CALL DATATR (ARRAY, TOP)
170    CALL CLOCK ( 1, '$DAT' )
      GO TO 10
C*
C*      GO HERE FOR SIGEXT MODULE
C*
175    CONTINUE
*** SIGEXT
176    CALL CLOCK ( 1, '$SIG' )
      GO TO 10
C
C      GO HERE FOR TRSTAT
C
180    CONTINUE
      CALL TRSTAT (ARRAY, TOP)
190    CALL CLOCK ( 1, '$TRS' )
      GO TO 10
C
C      GO HERE FOR NDHIST
C
200    CONTINUE
      CALL NDHIST (ARRAY, TOP)
210    CALL CLOCK ( 1, '$NDH' )
      GO TO 10
C
C      GO HERE FOR SCTRPL
C
220    CONTINUE
      CALL SCTRPL (ARRAY, TOP)
230    CALL CLOCK ( 1, '$SCT' )
      GO TO 10
C
C
C      GO HERE FOR DOTDATA
C
240    CONTINUE
      CALL DOTDAT (ARRAY, TOP)
250    CALL CLOCK ( 1, '$DOT' )
      GO TO 10
C
C      GO HERE FOR LABEL
C
260    CONTINUE
      CALL LABEL (ARRAY, TOP)
270    CALL CLOCK ( 1, '$LAB' )
      GO TO 10
C
C      GO HERE FOR EQUI-PROBABLE BLOCKS CLASSIFIER
C
280    CONTINUE
      CALL EQUIPRH (ARRAY, TOP)
285    CALL CLOCK ( 1, '$EQU' )
      GO TO 10
C
C      GO HERE FOR MULTI-TEMPORAL CLASSIFIER
C
290    CONTINUE
      CALL MULRAY (ARRAY, TOP)
295    CALL CLOCK ( 1, '$MUL' )
      GO TO 10
C
C      GO HERE FOR GROUND TRUTH TO MAPFIL
C
300    CONTINUE
      CALL GTTCN (ARRAY, TOP)
305    CALL CLOCK ( 1, '$GTT' )
      GO TO 10

```

```

MON01530
MON01540
MON01550
MON01560
MON01570
MON01580
MON01590
MON01600
MON01610
MON01620
MON01630
MON01640
MON01650
MON01660
MON01670
MON01680
MON01690
MON01700
MON01710
MON01720
MON01730
MON01740
MON01750
MON01760
MON01770
MON01780
MON01790
MON01800
MON01810
MON01820
MON01830
MON01840
MON01850
MON01860
MON01870
MON01880
MON01890
MON01900
MON01910
MON01920
MON01930
MON01940
MON01950
MON01960
MON01970
MON01980
MON01990
MON02000
MON02010
MON02020
MON02030
MON02040
MON02050
MON02060
MON02070
MON02080
MON02090
MON02100
MON02110
MON02120
MON02130
MON02140
MON02150
MON02160
MON02170
MON02180
MON02190
MON02200
MON02210
MON02220
MON02230
MON02240
MON02250
MON02260
MON02270
MON02280

```

FILE MONPAC

```

C
C      GO HERE FOTR IMAGE DATA MERGE
C 310  CONTINUE
      CALL DAMKG(ARRAY, TOP)
315    CALL CLOCK ( 1, 'SDAM' )
      GO TO 10
C
C      GO HERE FOR AMOEBA
C 320  CONTINUE
      CALL AMOEBA(ARRAY, TOP)
325    CALL CLOCK ( 1, 'SAMO' )
      GO TO 10
C
C      GO HERE FOR CLASY
C 330  CONTINUE
      CALL CLASY(ARRAY, TOP)
335    CALL CLOCK ( 1, 'SCLS' )
      GO TO 10
C
C      GO HERE FOR TESTSP
C 340  CONTINUE
      CALL TESTSP(ARRAY, TOP)
345    CALL CLOCK ( 1, 'STES' )
      GO TO 10
C
C      GO HERE FOR GROUND TRUTH DOT UNLOAD
C 350  CONTINUE
      CALL GTDDM(ARRAY, TOP)
355    CALL CLOCK ( 1, 'SGTD' )
      GO TO 10
C
C      GO HERE FOR PCG
C 360  CONTINUE
      CALL PGSTAT(ARRAY, TOP)
      CALL CLOCK(1, 'PCG')
      GO TO 10
C
C      GO HERE TO EXIT
C 370  IF (NOFILE .GT. 0) REWIND MAPUNT
      END

```

MON02290
 MON02300
 MON02310
 MON02320
 MON02330
 MON02340
 MON02350
 MON02360
 MON02370
 MON02380
 MON02390
 MON02400
 MON02410
 MON02420
 MON02430
 MON02440
 MON02450
 MON02460
 MON02470
 MON02480
 MON02490
 MON02500
 MON02510
 MON02520
 MON02530
 MON02540
 MON02550
 MON02560
 MON02570
 MON02580
 MON02590
 MON02600
 MON02610
 MON02620
 MON02630
 MON02640
 MON02650
 MON02660
 MON02670
 MON02680
 MON02690
 MON02700
 MON02710
 MON02720
 MON02730
 MON02740

4. MSCAN

FILE MSCAN

```

SUBROUTINE MSCAN(MGO,DBUG)
  IMPLICIT INTEGER (A-H,O-Z)
  DIMENSION CONTAM(24),COMENT(15),DATE(3)
  1, HED1(15), HED2(15),ACARD(20)
  -----
  CALL.. CALL MSCAN(MGO,DBUG)
  CONTINUE
  ARGS.. JGO - PROCESSOR PTR
         DBUG - DEBUG KEY -1 FIRST ENTRY
                0 - INCLUDE FLASH
                1 - EXCLUDE FLASH

  PURPOSE.. ANALYZES ALL MONITOR CONTROL CARDS

  RETURNS.. JGO - 1 %STAT
                  2 %CLASS
                  3 %DISPLAY
                  4 %SELECT
                  5 %HIST
                  6 %ISOCLS
                  7 %GPMAY
                  8 %DATA-TRANSFORMATION
                  9 %SIG EXT
                 10 %TSTAT
                 11 %NDHIST
                 12 %SCTRPL
                 13 %DOTDAT
                 14 %LREL
                 15 %EQUIPROB BLOCKS
                 16 %MULTI-TEMPORAL BAYES CLASSIFIER
                 17 %GROUND TRUTH TO CLUSTER MAP
                 18 %DATA MERGING
                 19 %AMOEBA
        CONTINUE
                 20 %CLASY
                 21 %TESTSP
                 22 %GTDDM
                 23 %PCG
                 24 %EXIT

  -----
  EQUIVALENCE (HED1(1), HEAD(4)), (DATE(1), HEAD(22)),
  1 (HED2(1), HEAD(3)), (COMENT(1), HEAD(48))
  INCLUDE COMHKA.LIST
  COMMON/GLOBAL/HEAD(63),MARTAP,DATE,SAVTAP,HMFILE,HMKEY,
  * HISFIL,HISKEY,IRFORM,ERIPTR,ERPKEY,MAPUNT,NOFILE,
  * DRUMAD,DRMUS,PAGSIZ,DAIFIL,STAFIL,ASAV,ASAVFL
  * NHSTON,NHSTFI,SCTRON,MAPFIL
  * DOTUNT,DOTFIL,NCHPAS,TRNSFL,BMTRFL,HISTFL,PCHUNT,
  * CRDUNT,PRUNT,RANDIO
  C$END
  COMMON /TAPEND/ IUNIT,IFIRST,FSCAN,SAMEND,SAMINC,READY,NSCAN,
  * LINC,IO(20),OSL,LBUF(30),JREC(30),IBYTE(30),NBUFS,FILENO,
  * LINENO,LININC,OSAMP,NOCHAN,FORMAT
  COMMON/ISTOR/IO(250)
  DATA CONTAM /1-STAT,2-CLASS,3-DIS,4-SEL,5-HIS,6-ISO,7-SGRA,
  * 8-DAT,9-SIG,10-IRF,11-SNM,12-SCT,13-DOT,
  * 14-LREL,15-EQ,16-MULT,17-GT,18-PCG,19-EXIT/
  * 20-CLAS,21-TEST,22-GTDD,23-PCG,24-EXIT/
  DATA FF /1-FORM,2-LFORM,3-LTHREE,4-LFOUR,5-4/
  DATA LHCD/'H',HCD/'H',EBCD/'E'
  DIMENSION CARD(62)
  -----
  INIZ
  -----

```


5. COMMON BLOCKS AND BLOCK DATA

The common block listings are given in this section. For specific descriptions, definitions of the parameters, and processor and subprogram interfaces, see section 5 of volume III of this user guide. The common block listings are given here in alphabetical order, as they appear in volume III.

COMMON/ARRAY(10600)

COMMON/BESTKN/KPPPTS(60),IPRIOR,KBEST,NCPASS

COMMON/BMTRX/BMATRX(450)

COMMON/CLASS/APRFLG,BMCOMB,BMFEAT,BMFLG,NOCAT,THIJ1,IDATA1,NFILE,STATKY,
CATNAM(60),CLSSYM(60),CON(60),DET(60),FLDESC,FLDINF(6),KCLSNA(60),
NOCTCL(60),SUBCAT(60),NOCHAN,CHNVEC(30)

COMMON/DISPL/CATFLG,CATNAM(61),CLSNAM(61),SUBNAM(61),SUBNO(60),SUBCAT(60),
CLSSUB(60),NOMAP,TOTVT3,NOSUB3,PCFDKY,TSTKEY,TRNKEY,THRSKY,STATKY,
EMPTRS,THRSVA,PLTKEY,BMFLG,BMCOMB,BMFEAT,CDATE(2),FLDSV2,FIELD2,VERTX2,
FLDSV3,FIELD3,VERTX3,PCTID3,THRES(60),SYMMTX(66),HIGH(60),CON(60),FLDKEY,
NOFLD2,NOFLD3,NOFET2,FETVC2(30),NOSUB2,NOTRFD,TOTVT2,NOCLS2,KATNO(60),
NOCAT,FILTER,MAPFMT,DESKEY,DESUNI,DESOTH,CROP,ACROP,AOTHER,ATOTAL,SITE(6),
ANALYS(5),CAM(15),CRPKEY,KEPPTS(60),DOTKEY,DOTERR

COMMON/DOTVEC/TYPE,CATNAM(60),NOCAT,TOTVEC,FLDINF(6),PRTKEY,SIZE,LACIE

COMMON/DVNBK/DFDK,CAYMIN,FII,CCAY,I1D,I1DMEN,ITT,ICNT,N

COMMON/FNTDUM/ITT,ICYCLE

COMMON/FSL/CFAC,TOTMSR,SEPMSR,PRCKEY,CRIKEY,INCFET,INCVEC(30),ICOUNT,SETWGT,
EVALBF(100),FETVC4(30),NOFET4,VARSZ4,CORBAS,DTAB4,WGHS14,BESTVC(10),
DIVSIZ,STATKY,ADRES,ADRESP,ADRESF,ADRS1,ADRS2

COMMON/GLOBAL/HEAD(63),MAPTAP,DATAPE,SAVTAP,BMFILE,BMKEY,HISFIL,HISKEY,
TRFORM,ERIPTP,ERPKEY,MAPUNT,NOFILE,DRUMAD,DRMWDS,PAGSIZ,DATFIL,STAFIL,
ASAV,ASAVFL,NHSTUN,NHSTFI,SCTRUN,MAPFIL,DOTUNT,DOTFIL,NCHPAS,TRNSFL,
BMTRFL,HISTFL,PCHUNT,CRDUNT,PRTUNT,RANDIO

COMMON/GRCBLK/MAXFET,NOFEAT,NOFET2,FETVEC(30),FETVC2(30),FLDINF(6),INFMT,
FILESV,NOHIST,HISVEC(30),NOFLD,FLDPTS,XSIZ,XLOW,XHGH,YSIZ

COMMON/GBK/NRDR,NPRT,PRTKEY,VLB(6),GTRDU,GTRDF,GTWRU,GTWRF,GTNOF

COMMON/HISTOR/HF

COMMON/IDSTOR/IDD(250)

COMMON/IDWORD/IDWORD(1000)

COMMON/INFORM/NOCLS2,NOSUB2,NOFET2,VARSZ2,TOTVT2,NOFLD2,AVAR2,COVAR2,CLSID2,
SUBNO2,SUBDS2,FLDSV2,VERTX2,FETVC2(30),SUBVC2(75),SUBPTR(75),CLSV2(60),
KEPPTS(60),NOGRP,GRPNAM(60),GRPDEX(61),GRPCHK(61),GROUPS(124)

COMMON/ISOLNK/SUNANG(8),ISUNT,ISUNC,SMSTR,SMSTP,SMINC,LINSKP

COMMON/LABS/NOCAT,CATNAM(60),NOCL2,CLSNM2(60),NOCAT2,CATNM2(60),SUBRAY(120),
PTR(60),CATPTR(250),CATDOT(500),DOTVEC(250),COND,MIX,PROC,MAPKEY,
DOTKEY,STATKY,SUNANG,T,NEARST,DIST,NOFEAT,FETVEC(30),OMAPUN,OMAPFI,
OSAVTP,OSTAFI,NOSUN,ANGLE(8),SIZE,TOTDT2,FLDINF(6),CLSSYM(62),STADRS,
MEANAD,TABADR,MAPADR,SUNCOR(30),ODOTUN,ODOTFI,MANSTA,MANDOT,DSPUNT,
DSPFIL,DSPKEY,PRNSTS,PRNDOT,FLDNAM,VERTEX(22),NOVRT,NSUN,ANGLES(8),
TOTDT3,FLDADR,VTXADR

COMMON/LISTMM/NPGA(3,2),NAMPGA(209,3,2),LINPGA(209,3,2),SAMPGA(209,3,2),
DOTLAB(209,4,2),VPGA(3),IPGA

COMMON/MRGDAT/IMOPT,ISOPT,NUMFIL,IDATTP(6),IDATFL(6),NOFEAT,NFEAT(6),
FETVEC(30,6),ISUN(8,6),SUNCOR(30),FLDINF(6,6),NOSAMP,NOLINE,NSS(6),
NACROS,NLINES(6),LINPTR(7),LINES(600),FORMM

COMMON/NDIM/NCLRCH,CLRVEC(30),MAXVEC,MAPKEY,CLASS,SUBCLS,FIELD,MEANSW,
NOVEC,FLDINF(6),SIZE,TOTMNS,CNTR1,CNTR2,ID1,ID2,COLOR1,COLOR2,BUFLEN,
ID3,COLOR3,NODUMP,IDATA1,TOTVEC

COMMON/PASS/STOP,LNCAT,NMIN,KRN,STDMAX,DLMIN,SEP,MAP,SPTRIG,IRD,KPTS,NOPTS,
PUNCH,ICHN,CHNTHS,ICHAIN(62),NWDS,IBEGIN,BEGIN1,BEGIN2,BEGIN3,CLSNAM,
NOFLD,IPT,TOTWRD,TOTPTS,NCLASS,NOCLS,TOTSUB,TOTFLD,TOTVRT,NOCL,NVRT,
NXTCLS,NOFEAT,MAXCLS,FETVEC(30),SYMMTX(62),VARSIZ,STATKY,ISOKEY,MAPFMT,
MAPKEY,SEQUEN(20),PERCEN,SIMERP,IORDER,INUNIT,INFILE,INITM,PMIN,SUBVEC(62),
NOSUB2,CHNVC(30),NOCHAN,ERCOMP,NOSEQ,MEANDU,MEANDU,SYMDU,SYMDU,ITRIGO,
ITRIGU,DOFLAG,DUFLAG,DODU,STDOTS(60),NSDOTS,SUNCOR(30),LLNCAT,DVERT(250,2),
DRECT(60,2),DVPNT(11,2),IDCNT(2),NDOU(2),MXFET1,MAXPOP

COMMON/PASSA/NOFET1,FTVEC1(30)

COMMON/PASSB/NOCLS,NOSUB,NOFEAT,NOFLD,TOTVRT,FETVEC(30),FLDSV1,CLSID1

COMMON/SCRACH/SCR1(2000),SCR2(10500)

COMMON/SCTTER/RSCALE,XYSCALE,CLRVEC(30),NCLRCH,CLRKEY,LOG,FREQ,XMAX,YMAX,
XMIN,YMIN,BCKGND,XHI,XLO,YLO,XSIZ,YHI,YSIZ,NBINS,SYMMTX(32),BMATRX(60),
BVEC(30),NB:CHN,NOFEAT,SCALKY,MENADR,FLDADR,PNTADR,IDADR,NC,BMFEAT,
BMCOMB,NOVEC,TOTMNS,SIZE,DRMID,DRMID1,DRMCLR,DRMCRI,DRMTNS,DRMTN1,
DRMCNT,DRMCT1,DRMVEC,DRMVC1,VECTR1,DATA1,NVEC,NOREAD,LREAD,DRMPTR,
DRMPT1,FETVEC(16),DRMPLT,CSCALE,NOSUB

COMMON/STBASE/SUBSV1,SUBMN1,SUBVR1,SUBSD1,SUBCL1,SAVER1,HSTAL1,SPEC1,COVAR1,
AVAR1,CLSID1,FLMEN1,FLVAR1,HFTAL1,FLDSV1

COMMON/STCBLK/MAXFET,MAXCLS,MAXFLD,NOFEAT,NOFET2,VARSIZ,NOSPEC,NOHIST,
SPCBAS,IBLOCK(30),FETVEC(30),FETVC2(30),HISVEC(30),NOFLD,NOCLS,
FLDINF(6),FLDPTS,CLSPTS,XSIZ,XHGH,XLOW,YSIZ

COMMON/TAPERD/IUNIT,IFRST,FSCAN,SAMEND,SAMINC,READY,NSCAN,LINC,ID(200),
DSL,LBUF(30),JREC(30),IBYTE(30),NBUFS,FILENO,LINEND,LININC,NSAMP,
NOCHAN,FORMT

COMMON/TR/TRNS1(256),TRNS2(26),TRNS3(26),TY(11,19)

COMMON/TRBLCK/OUTFMT,NOFEAT,FLDINF(6),FETVEC(30)

COMMON/WRTAP/ICOUNT,FORMT,UNIT,VARBL(600),IREMD

6. HIST PROCESSOR

FILE: HIST

C	SUBROUTINE HIST (ARRAY, TOP)	HIS000010
C	IMPLICIT INTEGER (F-T)	HIS000020
C	DIMENSION ARRAY (1)	HIS000030
C	//HIST	HIS000040
C	-----	HIS000050
C	-----	HIS000060
C	CALL. CALL HIST (ARRAY, TOP)	HIS000070
C	REQUIRES. NO COMMON BLOCKS	HIS000080
C	ROUTINES SETUP5	HIS000090
C	HISTGM	HIS000100
C	PURPOSE. COORDINATES THE LOGICAL STEPS FOR HISTOGRAMMING	HIS000110
C	DATA	HIS000120
C	RETURNS. NONE	HIS000130
C	-----	HIS000140
C	-----	HIS000150
C	CALL SETUP5 TO READ IN CONTROL CARDS	HIS000160
C	CALL SETUP5 (FILHS, FLOTL, TOTTL, TOP)	HIS000170
C	CALL HISTGM (ARRAY (FILHS), ARRAY (FLOTL), ARRAY (TOTTL))	HIS000180
C	RETURN	HIS000190
C	END	HIS000200
		HIS000210
		HIS000220
		HIS000230
		HIS000240
		HIS000250
		HIS000260
		HIS000270
		HIS000280
		HIS000290
		HIS000300
		HIS000310

```

SUBROUTINE SETUP(FILMS,FLOTL,TOTLT,TOP)
IMPLICIT INTEGER(A-Z)
PURPOSE -- HEADS THE CONTROL CARDS FOR HIST PROCESSOR

INCLUDE COMMK3.LIST
INCLUDE COMMK4.LIST
INCLUDE COMMK6.LIST
COMMON /GRCHLK/MAXFET,NOFEAT,NOFET2,FETVEC(30),
      FETVC2(30),FLDINF(6),INfmt,FILESV,NOMHST,
      HISVEC(30),NOFLD,          FLDPTS
      *XSIZ,XLOW,XHIGH,Ysiz
DIMENSION HED1(15),HED2(15),DATE(3),COMENT(15)
EQUIVALENCE (HED1(1),HEAD(4)),(DATE(1),HEAD(22)),
      (HED2(1),HEAD(30)),(COMENT(1),HEAD(48))
COMMON/GLOBAL/HEAD(63),MAPTAP,DATAPF,SAVTAP,HMFILE,BMKEY,
      *HISFIL,HISKEY,TRFORM,FRIPTP,ERPKEY,MAPUNT,NOFILE,
      *DHUMAD,DRMWDS,PAGSIz,DATFIL,STAFIL,ASAV,ASAVFL
      *NMSTUN,NMSTFI,SCTPUN,MAFFIL
      *DOTUNT,DOTFIL,NCHPAS,TPNSFL,BMTRFL,HISTFL,PCHUNT,
      CRDUNT,PRUNT,RANDIO

DIMENSION CARD(62),ACARD(20)
DIMENSION EQUIVEC(2)
DIMENSION INVECT(9),NUMVEC(30),IPTVEC(30),IPT(2),CHAR(2)
      * , NUMVC1(30)

DATA INVECT/'CHAN','HED1','HED2','COMM','DATE',
      *,'DISP','EQN','SIZE','DATA'/
DATA EQUIVEC/1,'='/,
DATA BLANK/' ','CHAR/1,'='/,EQUAL/'='/,IBCD/'I'/
DATA URCD/'U'/',LRCD/'L'/',MRCD/'M'/
DATA YRCD/'Y'/',FRCD/'F'/',XRCD/'X'/

INfmt=2
XSIZ = 101
YSIZ = 15
XHIGH = 255
XLOW = 0
NOFEAT = 0
NOMHST = 0
CALL TDATE(DATE)
WRITE(6,630)

SETUP REREAD BUFFER

CALL REREAD(30,80)
NOW PUT THE NEXT CARD IN THE BUFFER
10 READ(21,15)(ACARD(I),I=1,20)
15 FORMAT(20A4)
WRITE(30,15)(ACARD(I),I=1,20)
REWIND 30
READ(30,480)CODE,CARD
REWIND 30
WRITE(6,550) CODE,CARD
COL = 0
DO 20 I=1,9
IF(CODE.EQ.INVECT(I)) GO TO (30,40,50,60,70,
      * 90,150,100,170).I
20 CONTINUE
WRITE(6,440) CODE,CARD
GO TO 10

CHANNEL CARD

30 J = NATCHR(CARD,COL)
IF(J.FQ.BLANK) GO TO 10
COL = COL + 1
NOFEAT = NUMBER(CARD,COL,NUMVEC,NOFEAT)
CALL SORT(NOFEAT,IPT,NUMVEC,IPTVEC)
KA = IPT(1)
DO 35 I=1,NOFEAT
FETVEC(I) = NUMVEC(KA)
35 KA = IPTVEC(KA)
GO TO 10

HED1 CARD

```

FILE: SETUP5

```

40 READ(30,500)HED1
REWIND 30
GO TO 10
C
C HED2 CARD
50 READ(30,500)HED2
REWIND 30
GO TO 10
C
C COMMENT CARD
60 READ(30,500) COMENT
REWIND 30
GO TO 10
C
C DATE CARD
70 READ(30,510) DATE
REWIND 30
GO TO 10
C
C DISPLAY CARD
90 J = NXTCHR(CARD,COL)
IF (J.EQ.BLANK) GO TO 10
COL = COL + 1
NOHIST = NUMBER(CARD,COL,NUMVC1,NOHIST)
GO TO 10
C
C SIZE CARD
97 COL = COL - 1
100 J = NXTCHR(CARD,COL)
IF (J.EQ.BLANK) GO TO 10
IF (J.EQ.YRCD) GO TO 130
IF (J.EQ.YRCD) GO TO 140
GO TO 120
130 M = FIND12(CARD,COL,CHAR)
IF (CHAR(M).NE.EQUAL) GO TO 120
M = NUMBER(CARD,COL,NUMVEC,0)
YSIZ = NUMVEC(1)
GO TO 97
140 J = NXTCHR(CARD,COL)
M = FIND12(CARD,COL,CHAR)
IF (CHAR(M).NE.EQUAL) GO TO 120
M = NUMBER(CARD,COL,NUMVEC,0)
IF (J.EQ.HRCD) XHGH = NUMVEC(1)
IF (J.EQ.HRCD) GO TO 97
IF (J.EQ.LRCD) GO TO 120
XLOW = NUMVEC(1)
GO TO 97
C
C DATAFILE POSITIONING CARD
1701 M=NXTCHR(CARD,COL)
IF(M.EQ.BLANK) GO TO 10
IF(M.EQ.URCD) GO TO 1702
IF(M.EQ.FRCD) GO TO 1703
1705 WRITE(6,1705)
1704 FORMAT(' ERROR ON DATA FILE CARD ')
GO TO 10
1702 J=FIND12(CARD,COL,EQUVEC)
IF(J.EQ.-1) GO TO 1705
M=NUMBER(CARD,COL,DATAPE,ZERO)
COL=COL-1
GO TO 1701
1703 J=FIND12(CARD,COL,EQUVEC)
IF(J.EQ.-1) GO TO 1705
FILNO=NUMBER(CARD,COL,DATAFIL,FILNO)
DATAFIL=DATAFIL-1
COL=COL-1
GO TO 1701
C
C *END* CARD
150 CONTINUE
IF(NOHIST.EQ.0) GO TO 1
C

```

```

SET00800
SET00810
SET00820
SET00830
SET00840
SET00850
SET00860
SET00870
SET00880
SET00890
SET00900
SET00910
SET00920
SET00930
SET00940
SET00950
SET00960
SET00970
SET00980
SET00990
SET01000
SET01010
SET01020
SET01030
SET01040
SET01050
SET01060
SET01070
SET01080
SET01090
SET01100
SET01110
SET01120
SET01130
SET01140
SET01150
SET01160
SET01170
SET01180
SET01190
SET01200
SET01210
SET01220
SET01230
SET01240
SET01250
SET01260
SET01270
SET01280
SET01290
SET01300
SET01310
SET01320
SET01330
SET01340
SET01350
SET01360
SET01370
SET01380
SET01390
SET01400
SET01410
SET01420
SET01430
SET01440
SET01450
SET01460
SET01470
SET01480
SET01490
SET01500
SET01510
SET01520
SET01530
SET01540
SET01550
SET01560
SET01570
SET01580

```

FILE: SETUP

```

C      IS DISPLAY A SURSET OF CHANNEL CARD
C      DO 140 J=1,NOHIST
      DO 155 I=1,NOFEAT
      IF (NUMVC(I) .EQ. FETVEC(I)) GO TO 160
155  CONTINUE
      WRITE(6,560) NUMVC(I)
      NUMVC(I) = 0
160  CONTINUE
      CALL SORT(NOHIST,IPT,NUMVC,IPTVEC)
      KA = IPT(1)
      DO 165 I=1,NOHIST
      HISVEC(I) = NUMVC(KA)
165  KA = IPTVEC(KA)
      CONTINUE
C      CHECKING XHIGH AND XLOW
C      IF((XHIGH - XLOW) .GE. 100) GO TO 170
      XLOW = XHIGH - 100
      IF (XLOW .LT. 0) XHIGH = 100
      IF (XLOW .LT. 0) XLOW = 0
      WRITE(6,570) XHIGH,XLOW
170  CONTINUE
C      BASES FOR ARRAY
C      FILMS = 1
      FLOTL = NOFEAT*256 + FILMS
      TOTTL = NOHIST*XSIZ + FLOTL
      TOTPTS = NOHIST*XSIZ + TOTTL - 1
      IF(TOTPTS .LE. TOP) RETURN
C      RESET NO OF CHANNELS TO BE DISPLAYED AND RECALCULATE ADDRESS TOTTL
C      NOHIST1 = NOHIST
      NOHIST = (TOP - (FLOTL-1)) / (2*XSIZ)
      TOTTL = NOHIST*XSIZ + FLOTL
      NOHIST = NOHIST + 1
C      ZERO OUT CHANNELS THAT ARE NOT TO BE PLOTTED
C      DO 200 I=NOHIST,NOHIST1
200  HISVEC(I) = 0
      WRITE(6,580) NOHIST
450  FORMAT(/ ' TOO MANY CHANNELS ARE BEING HISTOGRAMMED AND PLOTTED. ' /
      * ' NO. OF CHANNELS PLOTTED WAS RESET TO ',15)
      RETURN
480  FORMAT(A4,6X,62A1)
490  FORMAT(' INVALID CARD -- IGNORED' / T5,A4,6X,62A1)
500  FORMAT(10X,15A4)
510  FORMAT(10X,3A4)
570  FORMAT(' XHIGH - XLOW WAS LESS THAN 100. XHIGH WAS RESET TO ',13,
      * ' 2X. ' OR XLOW WAS RESET TO ',13)
550  FORMAT(5X,A4,6X,62A1)
560  FORMAT(' CHANNEL ',12, ' IS NOT A SURSET OF THE CHANNELS GIVEN ON C
      * ' CHANNEL CARD ' )
630  FORMAT(/ ' INPUT SUMMARY ' )
120  WRITE(6,640) CODE,CARD
640  FORMAT(/ ' 1X,A4,6X,62A1 / ' BAD SUPERVISOR CONTROL CARD ' / )
      GO TO 10
      END

```

SET01500
 SET01600
 SET01610
 SET01620
 SET01630
 SET01640
 SET01650
 SET01660
 SET01670
 SET01680
 SET01690
 SET01700
 SET01710
 SET01720
 SET01730
 SET01740
 SET01750
 SET01760
 SET01770
 SET01780
 SET01790
 SET01800
 SET01810
 SET01820
 SET01830
 SET01840
 SET01850
 SET01860
 SET01870
 SET01880
 SET01890
 SET01900
 SET01910
 SET01920
 SET01930
 SET01940
 SET01950
 SET01960
 SET01970
 SET01980
 SET01990
 SET02000
 SET02010
 SET02020
 SET02030
 SET02040
 SET02050
 SET02060
 SET02070
 SET02080
 SET02090
 SET02100
 SET02110
 SET02120
 SET02130
 SET02140
 SET02150
 SET02160
 SET02170
 SET02180
 SET02190
 SET02200

FILE: SORT

```

SUBROUTINE SORT(IA,IPT,NUMVEC,IPTVEC)
.....C
CALL      -- SORT(IA,IPT,NUMVEC,IPTVEC)
          IA      -- NO. OF ELEMENTS TO BE SORTED IN ASCEN CI
                  DING ORDER
          IPT      -- CONTAINS REG. AND ENDING POINTER FOR
                  IPTVEC ARRAY
          NUMVEC   -- ARRAY CONTAINING ELEMENTS TO BE SORTED
          IPTVEC   -- ARRAY CONTAINING POINTERS FOR NUMVEC CI

          EXAMPLE:
                  IPT = 2
                  IPT = R
          SMALLEST NO. = NUMVEC(IPT(1))
          NEXT NO.    = NUMVEC(IPTVEC(IPT(1)))
          LAST NO.    = NUMVEC(IPTVEC(IPT(2)))
.....C

C      DIMENSION IPTVEC( 30 ),IPT(2),NUMVEC( 30 )

      ICT      = 0
      NOHST1= IA
      DO 60 I = 1,IA
      ICT      = ICT + 1
      IF ( ICT .GT. 1 )      GO TO 10
      IPT(1) = 1
      IPT(2) = 1
      IF ( ICT .EQ. IPT(1) ) GO TO 60
10  KA      = IPT(1)
      KB      = IPT(2)
      IF (NUMVEC(I) .EQ. 0) NOHST1 = NOHST1 - 1
      IF (NUMVEC(I) .EQ. 0)      GO TO 60
      IF (NUMVEC(I) .LE. NUMVEC(KA)) GO TO 30
      IF (NUMVEC(I) .GE. NUMVEC(KB)) GO TO 40
20  KB      = KA
      KA      = IPTVEC(KA)
      IF (NUMVEC(ICT) .LE. NUMVEC(KA)) GO TO 50
      GO TO 20
30  IPTVEC(ICT) = IPT(1)
      IPT(1)      = ICT
      KB      = IPT(2)
      GO TO 60
40  IPT(2)      = ICT
      IPTVEC(KB) = ICT
      GO TO 60
50  IPTVEC(ICT) = KA
      IPTVEC(KB) = ICT
60  CONTINUE
      IA = NOHST1
      RETURN
      END

```

7. GRAYMAP PROCESSOR

FILE: GRAYMP

C	SUBROUTINE GRAYMP(ARRAY, TOP)	GRA00010
C	IMPLICIT INTEGER(A-Z)	GRA00020
C	DIMENSION ARRAY(TOP), RINLEV(30, 16)	GRA00030
C	DIMENSION SYMBOL(16, 2)	GRA00040
C	INCLUDE COMPK3.LIST	GRA00050
C	INCLUDE COMPK4.LIST	GRA00060
C	INCLUDE COMPK6.LIST	GRA00070
C	COMMON /GRCHLK/ MAXFET, NOFEAT, NOFET2, FETVEC(30),	GRA00080
C	FETVC2(30), FLOINF(6), INFMT, FILESV, NOHIST,	GRA00090
C	HISVEC(30), NOFLO, FLDPTS	GRA00100
C	* XS17, XIOW, XHGH, YSIZ	
C	DIMENSION MED1(15), MED2(15), DATE(3), COMENT(15)	COM00010
C	EQUIVALENCE (MED1(1), HEAD(4)), (DATE(1), HEAD(22)),	COM00020
C	(MED2(1), HEAD(30)), (COMENT(1), HEAD(48))	COM00030
C	2 COMMON/GLOBAL/HEAD(63), MAPTAP, DATAPE, SAVTAP, RMFILE, RMKEY,	COM00040
C	HISFIL, HISKEY, TRFORM, ERIPTP, ERKKEY, MAPUNT, NOFILE,	COM00050
C	* DRUMAD, DRUMWS, PAGESZ, DATFIL, STAFIL, ASAV, ASAVFL	COM00060
C	* NHSTUN, NHSTFI, SCTRUN, MAPFIL	COM00070
C	* DOTUNT, DOTFIL, NCHPAS, TRNSFL, BMTFPL, HISTFL, PCHUNT,	COM00080
C	* CRDUNT, PRTUNT, RANDIO	COM00090
C	C\$END	COM00100
C	COMMON /HISTOR/HF	GRA00120
C	CALL SETUP6(ARRAY, HINCNT, RINLEV, NUMBIN, SYMBOL, SYMCNT, SYMDIM)	GRA00130
C	IF ((HINCNT.EQ.1).OR.(HISKEY.EQ.1)) GO TO 1	GRA00140
C	FILMS=1	GRA00150
C	FLDTL=4000	GRA00160
C	TOTTL=9000	GRA00170
C	HF=1	GRA00180
C	CALL HISTGM(ARRAY(FILMS), ARRAY(FLDTL), ARRAY(TOTTL))	GRA00190
C	CALL SETUP6(ARRAY, HINCNT, RINLEV, NUMBIN, SYMBOL, SYMCNT, SYMDIM)	GRA00200
C	1 CALL PICT(ARRAY, RINLEV, NUMBIN, SYMBOL, SYMCNT, SYMDIM)	GRA00210
C	HF=0	GRA00220
C	RETURN	GRA00230
C	END	GRA00240
		GRA00250
		GRA00260
		GRA00270

FILE: HEADNG

```

SUBROUTINE HEADNG(TYPE,FETNUM,RINLEV,NUMBIN,FLOINP,
* SYMOL,NSAMP,FET,SYMDIM,TCOL)
  IMPLICIT INTEGER(A-Z)
  HEA00010
  HEA00020
  HEA00030
  HEA00040
  HEA00050
  CALL.. CALL HEADNG(TYPE,FETNUM)
  HEA00060
  HEA00070
  HEA00080
  HEA00090
  HEA00100
  HEA00110
  HEA00120
  HEA00130
  HEA00140
  HEA00150
  HEA00160
  HEA00170
  HEA00180
  HEA00190
  HEA00200
  HEA00210
  HEA00220
  HEA00010
  HEA00020
  HEA00030
  HEA00040
  HEA00050
  HEA00060
  HEA00070
  HEA00080
  HEA00090
  HEA00100
  HEA00110
  HEA00120
  HEA00130
  HEA00140
  HEA00150
  HEA00160
  HEA00170
  HEA00180
  HEA00190
  HEA00200
  HEA00210
  HEA00220
  INCLUDE COMPK3.LIST
  INCLUDE COMPK4.LIST
  INCLUDE COMPK6.LIST
  COMMON /GRCHLK/MAXFET,NOFEAT,NOFET2,FETVEC(30),
  * FETVC2(30),FLDINF(6),INFMT,FILESV,NOHIST,
  * HISVEC(30),NOFLD, FLOPTS
  * XSIZ,XLOW,XHIGH,YSIZ
  DIMENSION HED1(15),HED2(15),DATE(3),COMENT(15)
  EQUIVALENCE (HED1(1),HEAD(4)),(DATE(1),HEAD(22)),
  2 (HED2(1),HEAD(30)),(COMENT(1),HEAD(48))
  COMMON/GLOBAL/HEAD(63),MAPTAP,DATAP,SAVTAP,BMFILE,BMKEY,
  * HISFIL,HISKEY,TRFORM,ERPTP,ERPKEY,MAPUNT,NOFILE,
  * DRUMAD,DRUMDS,PAGSIZ,DATFIL,STAFIL,ASAV,ASAVFL
  * ,NHSTUN,NHSTFI,SCTRUN,MAPFIL
  * ,DOTUNT,DOTFIL,NCHPAS,TRNSFL,BMTRFL,HISTFL,PCHUNT,
  * CRDUNT,PRUNT,RANDIO
  CSEND
  DIMENSION SYMOL(16,2),FET(1),FLDINP(7)
  DIMENSION TCOL(3,110),RINLEV(30,16)
  EQUIVALENCE (FLDINF(1),LINSTR),(FLDINF(2),LINEND),
  * (FLDINF(3),LININC),(FLDINF(4),SAMSTR),
  * (FLDINF(5),SAMEND),(FLDINF(6),SAMINC)
  IF (TYPE.EQ.2) GO TO 110
  IF (TCOL.EQ.1) GO TO 120
  J=FETNUM
  103 FIRST=0
  WRITE(6,101) FIRST,(BINLEV(J,MA),MA=1,NUMBIN)
  101 FORMAT(//12,16(3X,I3))
  DO 102 IZ=1,4
  WRITE(6,104)((SYMOL(MA,1),SYMOL(MA,1),SYMOL(MA,1),SYMOL(MA,1),
  * SYMOL(MA,1),SYMOL(MA,1)),MA=1,NUMBIN)
  104 FORMAT(2X,96A1)
  IF (SYMDIM.EQ.1) GO TO 102
  WRITE(6,105)((SYMOL(MA,2),SYMOL(MA,2),SYMOL(MA,2),SYMOL(MA,2),
  * SYMOL(MA,2),SYMOL(MA,2)),MA=1,NUMBIN)
  105 FORMAT(1H+,1X,96A1)
  102 CONTINUE
  120 CONTINUE
  CALL SETMRG(66,0,66)
  C CALCULATE AND PRINT SAMPLE NUMBERS
  JG=0
  SS=FLDINP(4)
  SE=FLDINP(5)
  DO 106 I=SS,SE,SAMINC
  JG=JG+1
  TCOL(1,JG)=I/100
  TCOL(2,JG)=MOD(I,100)/10
  TCOL(3,JG)=MOD(I,10)
  106 CONTINUE
  110 IF (TYPE.EQ.2) WRITE(6,111)
  DO 107 I=1,3
  107 WRITE(6,108) (ICOL(I,J),J=1,NSAMP)
  108 FORMAT(10X,110I1)
  WRITE(6,111)
  111 FORMAT(1H0)
  IF (TYPE.EQ.2) CALL SETMRG(66,4,62)
  RETURN
  END

```

FILE: PICT

```

SUBROUTINE PICT(BUF,BINLEV,NUMBIN,SYMBOL,SYMCNT,SYMDIM)
C
C      IMPLICIT INTEGER(A-Z)
C-----
CALL.. CALL PICT(IDATA)
C
C      ARGS.. IDATA - SCANNER DATA
C
C      ROUTINES HEADNG TAPHDR FLDINF LINERD LAREAD
C
C      PURPOSE.. PICTORIALY DISPLAYS FEATURES REQUESTED
C
C      RETURNS NONE
C-----
C
C      INCLUDE COMBK6.LIST
C      INCLUDE COMBK3.LIST
C      INCLUDE COMBK4.LIST
C      COMMON /GRCLBK/MAXFET,NOFFET1,NOFFET2,FETVEC(30),
C      *      FETVC2(30),FLDINF(6),INFMT,FILESV,NOHIST,
C      *      HISVEC(30),NOFLD,      FLOPTS
C      *      XSIZ,XLOW,XHIGH,YSIZ
C      DIMENSION HED1(15),HED2(15),DATE(3),COMENT(15)
C      EQUIVALENCE (HED1(1),HEAD(4)),(DATE(1),HEAD(22)),
C      *      (HED2(1),HEAD(30)),(COMENT(1),HEAD(48))
C      COMMON/GLOBAL/HEAD(63),MPTAP,DATE,SAVTAP,BMFILE,BMKEY,
C      *      HISFIL,HISKEY,TRFORM,ERIPTP,ERPKEY,MAPUNT,NOFILE,
C      *      DRUMAD,DRMWDS,PAGSIZ,DATEIL,STAFIL,ASAV,ASAVFL
C      *      NHSTUN,NHSTFI,SCRUN,MAFIL
C      *      DOTUNT,DOTFIL,NCHPAS,TRNSFL,BMTRFL,HISTFL,PCHUNT,
C      *      CRDUNT,PRUNT,RANDIO
C$END
C
C      EQUIVALENCE (FLDINF(1),LINSTR),(FLDINF(2),LINEND),
C      *      (FLDINF(3),LININC),(FLDINF(4),SAMSTR),
C      *      (FLDINF(5),SAMEND),(FLDINF(6),SAMINC)
C
C      DIMENSION BINLEV(30,16)
C      DIMENSION LCHAR(2,256),LIN(110),IDATA(12000),BUF(110,20),JSTAT(20)
C      DIMENSION SYMBOL(16,2),FET(1),FLDINP(6)
C      DIMENSION VERTCS(2,11),FL(8)
C
C      DATA BLANK/' '
C      DATA OPAR/'(',')','/','COMMA','/'
C
C      READ HEADER RECORD ON DATA TAPE
C      CALL TAPHDR(DATAP,DATEIL)
C      READ FIELD DEFINITION CARDS
C      20 RUNNO=LAREAD(FLDNAM,VERTCS,FLDINF,NC)
C      IF(RUNNO.LE.0) GO TO 1
C      CHECK TO SEE IF INFORMATION WILL FIT ON DRUM
C      NRUES=20
C      PTS=(FLDINF(5)-FLDINF(4))/FLDINF(6)+1
C      SPTS=PTS
C      LINES=(FLDINF(2)-FLDINF(1))/FLDINF(3)+1
C      NOFFET4=NOFFET2
C      27 NRUESZ=PTS*NOFFET4
C      TSAMP=LINES*NRUESZ
C      IF(TSAMP.LE.DRMWDS) GO TO 26
C      REDUCE NO. OF CHANNELS BY ONE,CHECK TO SEE IF FIELD WILL FIT
C      NOFFET4=NOFFET4-1
C      GO TO 27
C      26 IF(NOFFET2.NE.NOFFET4) WRITE(6,28) NOFFET4
C      29 FORMAT(' THE NO. OF CHANNELS FOR THIS FIELD HAS BEEN REDUCED TO',
C      *      '13.' SO ALL THE INFORMATION WILL FIT ON DRUM) '/' MAKE ANOTHER RUN
C      *      TO GRAYMAP OTHER CHANNELS')
C      CALL FLDINT(FLDINF,FETVC2,NOFFET4)
C      ADRES=DRUMAD
C      DO 29 I=1,LINES
C      CALL LINERD(IDATA,ENDTAP)
C      IF(ENDTAP.NE.0) GO TO 30
C      CALL RWRITE(ADRES,IDATA,NRUESZ,LSTAT)
C      31 IF(LSTAT.EQ.1) GO TO 31
C      ADRES=ADRES+NRUESZ
C      IF(ADRES.LE.DRMAD+DRMWDS) GO TO 29
C      WRITE(6,33)

```

FILE: PICT

```

33 FORMAT(' FIELD TOO LARGE,TERMINATING')
CALL CMERR
29 CONTINUE
30 CONTINUE
FLDINP(1)=FLDINF(1)
FLDINP(2)=FLDINF(2)
FLDINP(3)=FLDINF(3)
FLDINP(6)=FLDINF(6)
NFET=1
C FOR EACH FEATURE
DO 4 J=1,NOFET4
PTS=SPTS
NADRES=DRUMAD+(J-1)*SPTS
FLDINP(4)=0
FLDINP(5)=0
FET(1)=FETVC2(J)
WRITE(6,HEAD)
MM=NC-1
300 FORMAT('T24,'SAMPLE LINE      NO. OF'
*2X,'CHANNEL  FIELDNAME      INC      INC      VERTICES      VERTICES(SAMP
*1E,'LINE')
*7X,'13.7X,A4.7X,I4.2X,I4.7X,I2.7X,5(A1,I4,A1,I4,A1,2X)/
*51.5(A1,I4,A1,I4,A1,2X))
WRITE(6,300) FET(1),FLDNAM, SAMINC,LININC,MM,
*((OPAR,VERTCS (1,NM),COMMA,VERTCS(2,NM),CPAR),NM=1,MM)
TCOL=0
C SET UP VALUE-SYMBOL TABLE
DO 9 JD=1,SYMDIM
T2=BINLEV(J,1)+1
DO 8 I3=1,I2
8 LCHAR(JD,I3)=SYMBOL(1,JD)
DO 10 J1=2,NUMBIN
T1=BINLEV(J,J1-1)+2
T2=BINLEV(J,J1)+1
DO 10 I3=1,I2
10 LCHAR(JD,I3)=SYMBOL(J1,JD)
9 CONTINUE
FLDINP(4)=FLDINF(4)
AD=0
7 PPTS=PTS
ADRES=NADRES+AD
IF(PTS.LE.110) GO TO 5
FLDINP(5)=FLDINP(4)+109*FLDINP(6)
GO TO 6
5 FLDINP(5)=FLDINF(5)
6 PTS=PTS-110
C INITIALIZE TAPE READING FOR THIS FIELD
CALL FLDINT(FLDINP,FET,NFET)
NSAMP=(FLDINP(5)-FLDINP(4))/FLDINP(6)+1
IF(NSAMP.GE.110) GO TO 101
IF(NSAMP.EQ.PPTS) GO TO 101
WRITE(6,100) FLDINP(5)
PTS=0
FLDINF(4)=FLDINP(5)
100 FORMAT(' YOU HAVE ASKED FOR TOO MANY SAMPLES, THE LAST SAMPLE IS',
*15)
101 CONTINUE
LINES=(FLDINP(2)-FLDINP(1))/FLDINP(3)+1
TYPE=1
FETNUM=J
CALL HEADNG(TYPE,FETNUM,BINLEV,NUMBIN,FLDINP,SYMBOL,
* NSAMP,FET,SYMDIM,TCOL)
TCOL=1
LINCNT=0
C READ AND FILL 20 BUFFERS
DO 44 JA=1,NUFES
CALL RREAD(ADRES,RUF(1,JA),NSAMP,JSTAT(JA))
ADRES=ADRES+NUFESZ
LINCNT=LINCNT+1
44 CONTINUE
LINE=LINSTR
IRUF=1
C FINISHED READING
36 IF(JSTAT(IRUF).EQ.1) GO TO 36
DO 200 AM=1,NSAMP
200 LIN(MM)=HLANA
CALL FLDINT(VERTCS,NC,FL,LINE,NS,JJ)
DO 14 JD=1,SYMDIM
L=1

```

PIC00800
 PIC00810
 PIC00820
 PIC00830
 PIC00840
 PIC00850
 PIC00860
 PIC00870
 PIC00880
 PIC00890
 PIC00900
 PIC00910
 PIC00920
 PIC00930
 PIC00940
 PIC00950
 PIC00960
 PIC00970
 PIC00980
 PIC00990
 PIC01000
 PIC01010
 PIC01020
 PIC01030
 PIC01040
 PIC01050
 PIC01060
 PIC01070
 PIC01080
 PIC01090
 PIC01100
 PIC01110
 PIC01120
 PIC01130
 PIC01140
 PIC01150
 PIC01160
 PIC01170
 PIC01180
 PIC01190
 PIC01200
 PIC01210
 PIC01220
 PIC01230
 PIC01240
 PIC01250
 PIC01260
 PIC01270
 PIC01280
 PIC01290
 PIC01300
 PIC01310
 PIC01320
 PIC01330
 PIC01340
 PIC01350
 PIC01360
 PIC01370
 PIC01380
 PIC01390
 PIC01400
 PIC01410
 PIC01420
 PIC01430
 PIC01440
 PIC01450
 PIC01460
 PIC01470
 PIC01480
 PIC01490
 PIC01500
 PIC01510
 PIC01520
 PIC01530
 PIC01540
 PIC01550
 PIC01560
 PIC01570
 PIC01580

FILE: PICT

```

DO 201 IA=1,NSAMP
KA=(IA-1)*SAMINC+FLDINP(4)
DO 202 JK=L,JJ,2
IF(KA.LT.FL(JK)) GO TO 201
IF(KA.GT.FL(JK+1)) GO TO 203
IDT=RUF(IA,IRUF)+1
LIN(IA)=LCHAR(JD,IDT)
GO TO 201
203 L=L+2
IF(L.GT.JJ) GO TO 205
202 CONTINUE
201 CONTINUE
205 CONTINUE
IF(JD.NE.2) GO TO 21
WRITE(6,17) LINE,(LIN(JK),JK=1,NSAMP)
WRITE(4,102) LINE
102 FORMAT(1H+,T122,I5)
GO TO 14
17 FORMAT(1H+,I5,4X,110A1)
21 WRITE(6,15) LINE,(LIN(JK),JK=1,NSAMP)
15 FORMAT(16,4X,110A1)
14 CONTINUE
LINE=LINE+LININC
LINCNT=LINCNT+1
IF(LINCNT.GT.LINES) GO TO 37
CALL RREAD(ADRES,RUF(1,IBUF),NSAMP,JSTAT(IBUF))
ADRES=ADRES+NBUSZ
37 IBUF=IBUF+1
IF(IRUF.GT.NRUF) IRUF=1
IF(LINE.LE.LINEND) GO TO 36
TYPE=2
CALL HEADNG(TYPE,FETNUM,BINLEV,NUMBIN,FLDINP,SYMBOL,
* NSAMP,FET,SYNDIM,TCOL)
FLDINP(4)=FLDINP(6)+FLDINP(5)
AD=AD+110
IF(PTS.GT.0) GO TO 7
4 CONTINUE
GO TO 20
1 IF(RUNNO.EQ.0) RETURN
GO TO 20
END

```

PIC01590
 PIC01600
 PIC01610
 PIC01620
 PIC01630
 PIC01640
 PIC01650
 PIC01660
 PIC01670
 PIC01680
 PIC01690
 PIC01700
 PIC01710
 PIC01720
 PIC01730
 PIC01740
 PIC01750
 PIC01760
 PIC01770
 PIC01780
 PIC01790
 PIC01800
 PIC01810
 PIC01820
 PIC01830
 PIC01840
 PIC01850
 PIC01860
 PIC01870
 PIC01880
 PIC01890
 PIC01900
 PIC01910
 PIC01920
 PIC01930
 PIC01940
 PIC01950
 PIC01960
 PIC01970
 PIC01980
 PIC01990

ORIGINAL PAGE IS
 ON MICROFILME
 COPY

```

SUBROUTINE SETUP6(FILHIS,RINCNT,BINLEV,NUMBIN,SYMBOL,SYMNT,
* SYMDIM)
IMPLICIT INTEGER(A-Z)

CALL.. CALL SETUP6(FILHIS)
ARGS.. FILHIS - HISTOGRAM DATA ARRAY
REQUIRES COMMONS /INFORM/ /INFORS/
          /GLOBAL/ /HELP/
ROUTINES NATCHR FIND12 NUMBER
PURPOSE.. READS AND ANALYSES CONTROL CARDS FOR 'GRAYMAP' STEP
RETURNS.. SUPERVISOR INFORMATION

INCLUDE COMRK3.LIST
INCLUDE COMRK4.LIST
INCLUDE COMRK6.LIST
COMMON /GRCLBK/MAXFFT,NOFEAT,NOFET2,FETVEC(30),
* FETVC2(30),FLDINF(6),INFMT,FILESV,NOHIST,
* HISVEC(30),NOFLD, FLDPTS
* XSIZ,XLOW,XHIGH,YSIZ
DIMENSION HED1(15),HED2(15),DATE(3),COMENT(15)
EQUIVALENCE (HED1(1),HEAD(4)),(DATE(1),HEAD(22)),
2 (HED2(1),HEAD(30)),(COMENT(1),HEAD(48))
COMMON/GLOBAL/HEAD(63),MPTAP,DATEP,SAVTAP,RMFILE,RMKEY,
* HISFIL,HISKEY,TRFORM,ERIPPT,ERPKEY,MAPUNT,NOFILE,
* DRUMAD,DRMADS,PAGSIZ,DATEFIL,STAFIL,ASAV,ASAVL
* NHSTUN,NHSTF1,SCRUN,MAPFIL
* DOTUNT,DOTFIL,NCHPAS,TRNSFL,BMTRFL,HISTFL,PCHUNT,
* CROUT,PRUNT,RANDIO

COMMON /HISTOR/HF
EQUIVALENCE (FLDINF(1),LINSTR),(FLDINF(2),LINEND),
* (FLDINF(3),LININC),(FLDINF(4),SAMSTR),
* (FLDINF(5),SAMEND),(FLDINF(6),SAMINC)

DIMENSION FRVEC1(3),FRVEC2(3),NUMVEC(30)
DIMENSION SYMBOL(16,2),ACARD(20)
DIMENSION CINDEX(10),BINLEV(30,16),EQUVEC(2),
* SINVEC(3),CARD2(62),HGPT(30),FILHIS(NOFEAT,256),COMMA(2)

EQUIVALENCE (SINVEC(3),EQUAL)

DATA CINDEX/'CHAN','BINL','SYMB','FORM','HED1',
* 'HED2','COMM','DATE','END','DATA'/
DATA EQUVEC/1,'='/,
DATA SINVEC/2,'.',',','='/,CINMAX/10/,
* BLANK/' ',COMMA/1,',',FRVEC1/2,'I','O',
* FRVEC2/2,'U','L'/
DATA MRCD/'M',MRCD/'O',XRCO/'X',DLRBCD/'S',DOTBCD/'./',
1 FOLBCD/'-',MNSPCD/'-',SLHBCD/'/',STRBCD/'_',RLKBCD/' ',
2 FRCD/'F',URCD/'U'

IK=1
NOFET2=0
MAXFFT=30
IF(HF,F0.1) GO TO 80
SYMDIM=0
NUMBIN=0
RINCNT=0

SETUP REREAD BUFFER

CALL REREAD(30,80)
14 COL=0
PUT NEXT CARD IN BUFFER
READ(2),200 (ACARD(I),I=1,20)
200 FORMAT(20A4)
WRITE(30,200) (ACARD(I),I=1,20)
REWIND 30

```

FILE: SETUP6

```

      GO TO 14
C HED1 CARD
  8 READ(30,25) HED1
  REWIND 30
  GO TO 14
C HED2 CARD
  9 READ(30,25) HED2
  REWIND 30
  GO TO 14
  25 FORMAT(10X,15A4)
C COMMENT CARD
  10 READ(30,25) COMENT
  REWIND 30
  GO TO 14
C DATE CARD
  11 M=NXTCHR(CARD2,COL)
  IF(M.EQ.BLANK) GO TO 14
  IF(M.EQ.HRCD) GO TO 1702
  IF(M.EQ.FRCD) GO TO 1703
  1705 WRITE(6,1704)
  1704 FORMAT(' ERROR ON DATA FILE CARD *')
  GO TO 14
  1702 J=FNID12(CARD2,COL,EQUVEC)
  IF(J.EQ.-1) GO TO 1705
  M=NUMBER(CARD2,COL,DATAPE,ZERO)
  COL=COL-1
  GO TO 1701
  1703 J=FNID12(CARD2,COL,EQUVEC)
  IF(J.EQ.-1) GO TO 1705
  FILNO=NUMBER(CARD2,COL,DATAFIL,FILNO)
  DATAFIL=DATAFIL-1
  COL=COL-1
  GO TO 1701
C *FND* CARD
  12 IF(MSYM.EQ.1) GO TO 26
C DEFAULT SYMBOLS
  SYMCNT=10
  SYMDIM=2
  SYMBOL(1,1) = MRCO
  SYMBOL(1,2) = OLPHCO
  SYMBOL(2,1) = XRCO
  SYMBOL(2,2) = EQLRCO
  SYMBOL(3,1) = ORCO
  SYMBOL(3,2) = MNSRCO
  SYMBOL(4,1) = OHCO
  SYMBOL(4,2) = OHCO
  SYMBOL(5,1) = STRRCO
  SYMBOL(5,2) = MNSPCO
  SYMBOL(6,1) = EQLRCO
  SYMBOL(6,2) = EQLRCO
  SYMBOL(7,1) = DUTHCO
  SYMBOL(7,2) = DUTHCO
  SYMBOL(8,1) = MNSHCO
  SYMBOL(8,2) = MNSHCO
  SYMBOL(9,1) = SLHRCO
  SYMBOL(9,2) = SLHRCO
  SYMBOL(10,1) = BLKRCO
  SYMBOL(10,2) = BLKRCO
C CHECK TO SEE IF BINLEVELS INPUT
  26 IF(RINCNT.EQ.1) GO TO 27
  NUMHTN=SYMCNT
  IF(HISFIL.NE.13) STOP 2
C READ HISTOGRAM AND CALCULATE BINLEVELS
  IF(HISKEY.EQ.1) GO TO 80
  NOFEAT=NOFEAT2
  DO 81 IZ=1,NOFEAT2
    81 FETVEC(IZ)=FETVC2(IZ)
  RETURN
  80 READ(HISFIL) NOFEAT,(FETVEC(I),I=1,NOFEAT)
  READ(HISFIL) ((FILHIS(I,J),J=1,256),I=1,NOFEAT)
  REWIND HISFIL

```

SET01590
 SET01600
 SET01610
 SET01620
 SET01630
 SET01640
 SET01650
 SET01660
 SET01670
 SET01680
 SET01690
 SET01700
 SET01710
 SET01720
 SET01730
 SET01740
 SET01750
 SET01760
 SET01770
 SET01780
 SET01790
 SET01800
 SET01810
 SET01820
 SET01830
 SET01840
 SET01850
 SET01860
 SET01870
 SET01880
 SET01890
 SET01900
 SET01910
 SET01920
 SET01930
 SET01940
 SET01950
 SET01960
 SET01970
 SET01980
 SET01990
 SET02000
 SET02010
 SET02020
 SET02030
 SET02040
 SET02050
 SET02060
 SET02070
 SET02080
 SET02090
 SET02100
 SET02110
 SET02120
 SET02130
 SET02140
 SET02150
 SET02160
 SET02170
 SET02180
 SET02190
 SET02200
 SET02210
 SET02220
 SET02230
 SET02240
 SET02250
 SET02260
 SET02270
 SET02280
 SET02290
 SET02300
 SET02310
 SET02320
 SET02330
 SET02340
 SET02350
 SET02360
 SET02370

FILE: SETUP6

```

DO 44 I=1,NOFEAT
44 HGTPT(I)=0
DO 45 I=1,NOFEAT
DO 45 J=1,256
45 HGTPT(I)=HGTPT(I)+FILHIS(I,J)
C IF FEATURES CARD NOT INPUT,FETVC2 ARRAY= FETVEC ARRAY,NOFET2=NOFEAT
IF (NOFET2.NE.0) GO TO 60
NOFET2=NOFEAT
DO 61 I=1,NOFEAT
61 FETVC2(I)=FETVEC(I)
GO TO 100
60 IK=1
KT=NOFET2
102 DO 28 I=1,KT
DO 28 J=1,NOFEAT
C CHECK TO SEE IF FEATURES HISTOGRAMMED
IF (FETVC2(I).NE.FETVEC(J)) GO TO 29
GO TO 28
29 CONTINUE
WRITE(6,30) FETVC2(I)
30 FORMAT(1X,'THIS CHANNEL IS NOT HISTOGRAMMED',I3//)
NOFET2=NOFET2-1
IF (I.GT.KT) GO TO 100
IK=IK+1
DO 101 II=IK,KT
101 FETVC2(II-1)=FETVC2(II)
KT=NOFET2
IK=I
GO TO 102
28 CONTINUE
100 DO 103 I=1,NOFET2
DO 104 J=1,NOFEAT
IF (FETVC2(I).NE.FETVEC(J)) GO TO 104
GO TO 32
104 CONTINUE
C COMPUTE RINLEVELS FOR FEATURE
32 I=HGTPT(J)/NUMBIN
LNUM=NUMBIN-1
M=I
N=0
K=0
DO 33 JJ=1,LNUM
33 K=K+1
N=N+FILHIS(J,K)
IF (N.LT.M) GO TO 34
M=M+1
RINLEV(I,JJ)=K-1
IF (JJ.NE.1) GO TO 111
IF (K.EQ.1) RINLEV(I,1)=1
GO TO 33
111 IF (RINLEV(I,JJ).EQ.RINLEV(I,JJ-1)) RINLEV(I,JJ)=RINLEV(I,JJ-1)+1
33 CONTINUE
RINLEV(I,NUMBIN)=255
103 CONTINUE
27 CONTINUE
C PRINT OUT SETUP6 CARDS
WRITE(6,40) (FETVC2(I),I=1,NOFET2)
40 FORMAT(1X,'SUPERVISOR INFORMATION FOR GRAYMAP',1X,'CHANNELS GRAYMAP',
*PPEN APE',30I4)
C WRITE OUT SYMBOLS
WRITE(6,41) (SYMBOL(KZ,1),KZ=1,SYMCNT)
41 FORMAT(1X,SYMBOLS USED IN GRAYMAP ARE',16(2X,A1))
IF (SYMDIM.EQ.2) WRITE(6,42) (SYMBOL(KZ,2),KZ=1,SYMCNT)
42 FORMAT(1H*,27X,16(2X,A1))
RETURN
END

```

SET02380
 SET02390
 SET02400
 SET02410
 SET02420
 SET02430
 SET02440
 SET02450
 SET02460
 SET02470
 SET02480
 SET02490
 SET02500
 SET02510
 SET02520
 SET02530
 SET02540
 SET02550
 SET02560
 SET02570
 SET02580
 SET02590
 SET02600
 SET02610
 SET02620
 SET02630
 SET02640
 SET02650
 SET02660
 SET02670
 SET02680
 SET02690
 SET02700
 SET02710
 SET02720
 SET02730
 SET02740
 SET02750
 SET02760
 SET02770
 SET02780
 SET02790
 SET02800
 SET02810
 SET02820
 SET02830
 SET02840
 SET02850
 SET02860
 SET02870
 SET02880
 SET02890
 SET02900
 SET02910
 SET02920
 SET02930
 SET02940
 SET02950
 SET02960
 SET02970
 SET02980
 SET02990
 SET03000
 SET03010
 SET03020
 SET03030

8. STAT PROCESSOR

FILE: STAT

```

SUBROUTINE STAT(ARRAY, TOP)
PURPOSE.. COORDINATES THE VARIOUS ROUTINES
          FOR 'STATISTICS' STEP

IMPLICIT INTEGER (A-H, O-Z)
DOUBLE PRECISION ARRAY(1500)

DIMENSION KEPPTS(60)

INCLUDE COMHKB.LIST
STAT COMMON BLOCK
COMMON /STBASE/SUBSV1, SUBMN1, SURVP1, SURSD1, SUBCL1, SAVER1, MSTAL1,
*SPEC1, COVAR1, AVAR1, CLSID1, FLVAP1, FLVAP1, HFTAL1, FLOSV1
COMMON /STCHLK/ MAXFET, MAXCLS, MAXFLD, NOFEAT, NOFET2,
*VARSI7, NOSPEC, NOHIST, SPCBAS, INLOCK(30), FETVEC(30),
*FETVC2(30), HISVEC(30), NOFLD, NOCLS,
*FLDINF(5), FLDPTS, CLSPTS, XSIZ, XHGH, XLOW, YSIZ
COMMON BLOCK STBASE CONTAINS THE BASE ADDRESSES FOR THE STATISTICS
STORED IN 'ARRAY'

SUBSV1 - BASE ADDRESS IN 'ARRAY' FOR SUBCLASS INFORMATION
          (5*SUBNO)
          FOR EACH SUBCLASS INDEX
          1-CLASS NUMBER
          2-STARTING FIELD NUMBER
          3-ENDING FIELD NUMBER
          4-SUBCLASS NAME

SURMN1 - BASE ADDRESS IN 'ARRAY' FOR SUBCLASS MEANS
SURVP1 - BASE ADDRESS IN 'ARRAY' FOR SUBCLASS VARIANCES
SURSD1 - BASE ADDRESS IN 'ARRAY' FOR SUBCLASS NAMES
SUBCL1 - BASE ADDRESS IN 'ARRAY' FOR CLASS NUMBER
SAVER1 - BASE ADDRESS IN 'ARRAY' FOR TRAINING FIELD VERTICES
MSTAL1 - BASE ADDRESS IN 'ARRAY' FOR SUBCLASS HISTOGRAM TOTALS
SPEC1 - BASE ADDRESS IN 'ARRAY' FOR SPECTOGRAM INFORMATION
          (5*OSPEC)

COVAR1 - BASE ADDRESS IN 'ARRAY' FOR FIELD COVARIANCES
AVAR1 - BASE ADDRESS IN 'ARRAY' FOR FIELD MEANS
CLSID1 - BASE ADDRESS IN 'ARRAY' FOR CLASS NAMES
FLVAP1 - BASE ADDRESS IN 'ARRAY' FOR FIELD MEANS
FLVAP1 - BASE ADDRESS IN 'ARRAY' FOR FIELD VARIANCES
HFTAL1 - BASE ADDRESS IN 'ARRAY' FOR FIELD HISTOGRAM TOTALS
FLOSV1 - BASE ADDRESS IN 'ARRAY' FOR FIELD INFORMATION (10*MAXFLD)

CONTINUE
          FOR EACH FIELD INDEX
          1-FIELD NAME
          2-CLASS NUMBER
          3-SUBCLASS NUMBER
          4-NUMBER OF VERTICES
          5-STARTING LINE NUMBER
          6-ENDING LINE NUMBER
          7-STARTING SAMPLE NUMBER
          8-ENDING SAMPLE NUMBER
          9-LINE INCREMENT
          10-SAMPLE INCREMENT

CONTINUE
COMMON BLOCK STCHLK CONTAINS INFORMATION NEEDED BY ROUTINES IN STAT
MAXFET - MAXIMUM NUMBER OF CHANNELS
MAXCLS - MAXIMUM NUMBER OF CLASSES
MAXFLD - MAXIMUM NUMBER OF FIELDS
NOFEAT - NUMBER OF CHANNELS REQUESTED
VARSI7 - SIZE OF EACH COVARIANCE MATRIX (NOFEAT*(NOFEAT+1))/2
NOSPEC - NUMBER OF GROUPS OF SUBCLASSES TO PLOT
NOHIST - NUMBER OF CHANNELS TO HISTOGRAM
SPCBAS - MINIMUM RADIANCE VALUE ON Y AXIS OF SPECTRAL PLOT
INLOCK - ARRAY CONTAINING TRIGGERS TO CERTAIN OPTIONS IN STAT
FETVEC - ARRAY OF CHANNELS SELECTED
HISVEC - ARRAY OF CHANNELS TO HISTOGRAM
NOFLD - NUMBER OF FIELDS
NOCLS - NUMBER OF CLASSES
FLDINF - FIELD INFORMATION ARRAY
FLDPTS - NUMBER OF POINTS IN FIELD
CLSPTS - NUMBER OF POINTS IN CLASS
XSIZ - XHIGH-XLOW =101
XLOW - MINIMUM X VALUE TO BE HISTOGRAMMED =0
XHGH - MAXIMUM X VALUE TO BE HISTOGRAMMED =255
YSIZ - HEIGHT OF Y AXIS IN HISTOGRAM =15

CALL SETUP1(ARRAY, TOP, MAXSUB)
CALL LEARN(ARRAY(SPEC1), ARRAY(COVAR1), ARRAY(AVAR1))

```


FILE: STAT

```

      *      ARRAY(CLSID),ARRAY(SURSV),ARRAY(FLMEN),ARRAY(FLVAR) STA00800
      *      ARRAY(SURMN),ARRAY(SURVR),ARRAY(SUHSD),ARRAY(SUBCL) STA00810
      *      ARRAY(MFTAL),ARRAY(MSTAL),ARRAY(FLOSV),
      *      ARRAY(SAVER),KEPPTS,MAXSUB STA00830
2  WRITE (A,2) STA00840
   FORMAT(////////// 2X, '*** $STAT - COMPLETED ***' //) STA00850
   RETURN STA00860
   END STA00870
```

FILE: CLSSPC

```

      SURROUTINE CLSSPC(MEAN,SURSTD,IDVEC,PTRVEC,PLOT,
      *TITLE,NOFEAT,FETVEC,SPCHAS)
C      IMPLICIT INTEGER (A-H,O-Z)
      LOGICAL OVWFLG
      REAL MEAN(1),SURSTD(1),BIAS,INCR,MENI,DEVI
      DOUBLE PRECISION DEV(1),DMEAN(1)
C      DIMENSION PLOT(4,NOFEAT,49),PTRVEC(5)
      DIMENSION FETVEC(30)
      * SYMVEC(4),TAB(12),ERRLIN(7)
C      INCLUDE COMRKG.LIST
      COMMON/GLOBAL/HEAD(63),MAPTAP,DATAP,SAVTAP,RMFILE,RMKEY,
      * HISFIL,HISKEY,THFORM,FRPTP,ERPKEY,MAPUNT,NOFILE,
      * DRUMAD,DRMDS,PAGSIZ,DATFIL,STAFIL,ASAV,ASAVL
      * NHSTUN,NHSTFI,SCRUN,MAPFIL
      * DOTUNT,DOTFIL,NCHPAS,TRNSFL,BMTRFL,HISTFL,PCHUNT,
      * CROUT,PRUNT,RANDIO
C$FND
C      DATA SYMVEC/'S','I','O','E','P','I','/ , INCR/3.0/, NOLINE/49/,
      1 BLANK /' ' /
C      DATA TAB /'12','20','28','36','44','52',
      1 '60','68','76','84','92','99' /
C      DATA ERRLIN/'(1H)','T ' , ' ',1H(' ','A1',' ','1H'),' ','13) ' /
C      DATA DASH/'-----' /
C
C      SPEC INIZ
      WRITE(6,HEAD)
      WRITE(6,10021)
10021 FORMAT( / 34X,'SPECTRAL PLOT (MEAN,PLUS AND MINUS ONE STD. DEV.)
      1 FOR:' / )
      WRITE(6,1002) TITLE , (DASH,I=1,2)
1002 FORMAT(44X,'TRAINING SUBCLASS ',A4/44X,4A4,A2/)
      WRITE(6,20021)
      WRITE(6,5002) SYMVEC(2),IDVEC
      GO TO 250
C
C      -----
C      ENTRY FLDSPC(DMEAN,DEV,IDVEC,PTRVEC,PLOT,MEAN,SURSTD,FLDNAM,
      *NOFEAT,FETVEC,SPCHAS)
      WRITE(6,HEAD)
      WRITE(6,10021)
      WRITE(6,1004) IDVEC, (DASH,I=1,4)
1004 FORMAT(47X,'TRAINING FIELD ',A4/46X,4A6/)
      WRITE(6,20021)
20021 FORMAT( / 47X,'PLOT LEGEND:' / 47X,'-----' )
      WRITE(6,2002) SYMVEC(2),IDVEC
2002 FORMAT(60X,A1,' = FIELD ',A4)
C
      DO 100 J=1,NOFEAT
100 MEAN(J) = DMEAN(J)
      DO 300 I=1,NOFEAT
300 SURSTD(I)=DEV(I)
250 JPSTR = 2
      JPLST = 2
      JBAS = 1
C
C      COMBINED INIZ
C
300 BIAS = SPCHAS
      WRITE(6,4002)BLANK
      CNT = BIAS
      OVWFLW = 0
      OVWFLG = .TRUE.
      DO 350 J=1,NOFEAT
      DO 350 I=1,4
      DO 350 K=1,40
350 PLOT(I,J,K) = BLANK
      LOC=1
      ISTOP=NOFEAT
      IF (NOFEAT.GT.12) ISTOP=12

```

CLS00010
 CLS00020
 CLS00030
 CLS00040
 CLS00050
 CLS00060
 CLS00070
 CLS00080
 CLS00090
 CLS00100
 CLS00110
 CLS00120
 CLS00130
 CLS00140
 CLS00150
 CLS00160
 CLS00170
 CLS00180
 CLS00190
 CLS00200
 CLS00210
 CLS00220
 CLS00230
 CLS00240
 CLS00250
 CLS00260
 CLS00270
 CLS00280
 CLS00290
 CLS00300
 CLS00310
 CLS00320
 CLS00330
 CLS00340
 CLS00350
 CLS00360
 CLS00370
 CLS00380
 CLS00390
 CLS00400
 CLS00410
 CLS00420
 CLS00430
 CLS00440
 CLS00450
 CLS00460
 CLS00470
 CLS00480
 CLS00490
 CLS00500
 CLS00510
 CLS00520
 CLS00530
 CLS00540
 CLS00550
 CLS00560
 CLS00570
 CLS00580
 CLS00590
 CLS00600
 CLS00610
 CLS00620
 CLS00630
 CLS00640
 CLS00650
 CLS00660
 CLS00670
 CLS00680
 CLS00690
 CLS00700
 CLS00710
 CLS00720
 CLS00730
 CLS00740
 CLS00750
 CLS00760
 CLS00770
 CLS00780
 CLS00790

FILE: CLSSPC

```

C      SET UP 'PLOT' MATRIX
C
400  JKSV=0
      DO 690 JP=JPSTR,JPLOT
      MENBAS = (PTRVEC(JP-JBIAS)-1)*NOFEAT
      IF( OVRFLG) WHITE(6,4002)
4002  FORMAT(A4)
      OVRFLG = .FALSE.
      JK=JKSV
      JF = 0
      DO 500 JFEAT=LOC,ISTOP
      JK = JK+JFEAT
      MENI = MFAN(MENHAS+JFEAT)
      DEVI=SUHSTD(MENHAS+JFEAT)
      MENLOW = (MENI-DEVI-BIAS)/INCR+0.5
      MENHGH = (MENI+DEVI-BIAS)/INCR+0.5
      IF( MENLOW .GE. 1) GO TO 430
      OVRFLW = MENI-DEVI+0.5
      MENLOW = 1
430  IF(MENHGH .LE. NOLINE) GO TO 450
      OVRFLW = MENI+DEVI+0.5
      MENHGH = NOLINE
450  DO 490 J =MENLOW,MENHGH
490  PLOT(JP,JFEAT,J) = SYMVEC(JP)
      JF = JF + 1
      IF(OVRFLW .EQ. 0) GOTO 500
      JF = JF + 1 THIS STATEMENT MOVED UP ONE LINE *****
      FRRLIN(3)=TAR(JF)
      WRITE(6,ERHLINE) SYMVEC(JP),OVRFLW
      OVRFLG = .TRUE.
      OVRFLW = 0
500  CONTINUE
690  CONTINUE
C
C      PRINT OUT 'PLOT' MATRIX
C      WRITE(6,3004) CNT,CNT
3004  FORMAT(4X,13, 2X, 'I-----I', 12('-----I'), 2X, 13)
      CNT = CNT+INCR
700  DO 790 K= 1,NOLINE
      WRITE(6,7002)CNT,((PLOT(I,J,K),I=1,4),J=LOC,ISTOP)
      WRITE(6,7003)CNT
7002  FORMAT(4X,13,2X,'I',12(4X,4A1))
7003  FORMAT(' ',11X,'I',2X,13)
      CNT = CNT+INCR
790  CONTINUE
C
800  WRITE(6,3004) CNT, CNT
C
C      WRITE(6,2002) ( FETVEC(I), I=LOC,ISTOP)
8002  FORMAT( / 1X, 'CHANNEL NO.', 3X, 12, ( 11(6X,12) ) )
      WRITE(6,2003) (DASH,I=1,3)
8003  FORMAT(1X, 3A4 ///)
C
      IF(ISTOP.EQ.NOFEAT)RETURN
      CNT=BIAS
      JKSV=JK
      LOC=LOC+12
      ISTOP=ISTOP+12
      IF(ISTOP.GT.NOFEAT)ISTOP=NOFEAT
      GO TO 400
C
C
C
-----
C      ENTRY MULSPC(MFAN,SUHSTD,JOVEC,PTRVEC,PLOT,
      *NOFEAT,FETVEC,SPCRAS)
      DIMENSION JOVEC(1:1),RUF(4)
      WRITE(6,20021)
      JPSTR = 1
      JPLOT = PTRVEC(5)
      IF(JPLOT.NE. 1) GOTO 900
      JPSTR = 2
      JPLOT = 2
      JBIAS = 1
      WAT=PTRVEC(1)
      WRITE(6,5002) SYMVEC(2),JOVEC(1,WAT)
      GOTO 300
900  DO 22 JKL = JPSTR,JPLOT
      WAT=PTRVEC(JKL)

```

CLS00800
 CLS00810
 CLS00820
 CLS00830
 CLS00840
 CLS00850
 CLS00860
 CLS00870
 CLS00880
 CLS00890
 CLS00900
 CLS00910
 CLS00920
 CLS00930
 CLS00940
 CLS00950
 CLS00960
 CLS00970
 CLS00980
 CLS00990
 CLS01000
 CLS01010
 CLS01020
 CLS01030
 CLS01040
 CLS01050
 CLS01060
 CLS01070
 CLS01080
 CLS01090
 CLS01100
 CLS01110
 CLS01120
 CLS01130
 CLS01140
 CLS01150
 CLS01160
 CLS01170
 CLS01180
 CLS01190
 CLS01200
 CLS01210
 CLS01220
 CLS01230
 CLS01240
 CLS01250
 CLS01260
 CLS01270
 CLS01280
 CLS01290
 CLS01300
 CLS01310
 CLS01320
 CLS01330
 CLS01340
 CLS01350
 CLS01360
 CLS01370
 CLS01380
 CLS01390
 CLS01400
 CLS01410
 CLS01420
 CLS01430
 CLS01440
 CLS01450
 CLS01460
 CLS01470
 CLS01480
 CLS01490
 CLS01500
 CLS01510
 CLS01520
 CLS01530
 CLS01540
 CLS01550
 CLS01560
 CLS01570
 CLS01580

FILE: CLSSPC

```
22 RUF(JKL)=JDVEC(I,WAT)
WRITE(6,9002) (SYMVEC(I),RUF(I),I=JPSTR,JPLST)
9002 FORMAT(57X,A1,' = SUBCLASS ',A4)
      JRIAS = 0
      GO TO 300
      END
```

CLS01590
CLS01600
CLS01610
CLS01620
CLS01630
CLS01640

FILE: FLDCOV

```

SUBROUTINE FLDCOV(COR,DEV,MEAN,VAR,PTS,GO,FLDNAM,
*NOFEAT,MAXFFT,VARSIZ)
C
C   IMPLICIT INTEGER (A-H,O-Z)
C   INCLUDE COMPAK.LIST
COMMON/GLOBAL/MEAN(63),MAPTAP,DATAP,SAVTAP,BMFILE,BMKEY,
*   HISFIL,HISKEY,TRFORM,ERPTP,ERPKEY,MAPUNT,NOFILE,
*   DRUMAD,DRMADS,PAGSIZ,DATFIL,STAFIL,ASAV,ASAVFL
*   NMSTUN,NMSTFI,SCTRUN,MAPEIL
*   DOTUNT,DOTFIL,NCHPAS,TRNSFL,BMTRFL,HISTFL,PCHUNT,
*   CRDUNT,PTUNT,HANDIO
C$END
C
C   PURPOSE.. CALCULATES THE COVARIANCE AND CORRELATION MATRICES
C   FROM THE RAW DATA FOR THE FIELDS
C-----
C   DOUBLE PRECISION COR(VARSIZ),DEV(NOFEAT),MEAN(NOFEAT),VAR(VARSIZ)
C-----
C   DATA RCDT,0/ 12/ / , DASH/1-----1 /
C-----
C   IF ( GO .NE. 1) GO TO 20
C   WRITE(4,MEAN)
C   WRITE(4,10) NOFEAT
10  FORMAT(10// ' THE MEAN,STANDARD DEVIATION,COVARIANCE,AND CORRELATION
IN 11 12 ' CHANNELS) FOR: / / )
C   WRITE(4,11) FLDNAM, (DASH,1=1,3)
11  FORMAT(150,' TRAINING FIELD 'A4/T49.4A4/)
C
C   EQUATIONS :
C
C   COVAR(1,2) = 
$$\frac{1}{N-1} \sum_{J=1}^N \sum_{K=1}^N X_{1J} X_{2K} - U_1 U_2$$

C
C   MEAN(1) = 
$$\frac{1}{N} \sum_{I=1}^N X_1 = U_1$$

C
C   STDEV(2) = 
$$\sqrt{\text{COVAR}(2,2)}$$

C
20  JK = 0
PTS1 = PTS
PTS2 = PTS-1
IF (PTS2 .LT. 1) PTS2 = 1
N = NOFEAT
DO 40 JA=1,N
J = JA
DO 30 K=1,J
JK = JK+1
VAR(JK) = (VAR(JK)-MEAN(J)*MEAN(K)/PTS1)/PTS2
30  CONTINUE
DEV(J) = DSQRT(DABS(VAR(JK)))
40  CONTINUE
JK = 0
DO 50 J=1,NOFEAT
MEAN(J) = MEAN(J)/PTS1
DO 50 K=1,J
JK = JK+1
COR(JK) = 0.0
IF (DEV(K)*DEV(J).LT.1.0E-25) GO TO 50
COR(JK) = VAR(JK)/(DEV(J)*DEV(K))
50  CONTINUE

```

FILE: FLNCOV

C	50 CONTINUE	FLD00800
C	-----	FLD00810
C	IF (GO.EQ. 0) RETURN	FLD00820
	DO 70 LOC=1,NOFEAT,12	FLD00830
	STOP = LOC+11	FLD00840
	IF (STOP.GT. NOFEAT) STOP = NOFEAT	FLD00850
	WRITE (A,100) (MEAN(I), I=LOC,STOP)	FLD00860
	WRITE (A,100) (DASH,I=1,2)	FLD00870
	WRITE (A,110) (DEV(I), I=LOC,STOP)	FLD00880
	WRITE (A,100) (DASH,I=1,2)	FLD00890
	WRITE (A,120)	FLD00900
	70 CONTINUE	FLD00910
	100 FORMAT(10MEAN:1,12F9.2)	FLD00920
	1001 FORMAT(1X,A4,A2/)	FLD00930
	110 FORMAT(10ST DEV:1,12F9.2)	FLD00940
	120 FORMAT(10)	FLD00950
C	WRITE (A,130) (DASH, I=1,5)	FLD00960
	130 FORMAT(10) / ' COVARIANCE MATRIX' / 1X,A4,A2)	FLD00970
	CALL DWRITMX(VAR,NOFEAT,BCDTWO)	FLD00980
	IF (NOFEAT.LE. MAXFET) GO TO 140	FLD00990
	WRITE (A,HEAD)	FLD01000
C	140 WRITE (A,150) (DASH, I=1,5)	FLD01010
	150 FORMAT(10) // ' CORRELATION MATRIX' / 1X,A4)	FLD01020
	CALL DWRITMX(COR,NOFEAT,BCDTWO)	FLD01030
	RETURN	FLD01040
C	-----	FLD01050
C	-----	FLD01060
C	-----	FLD01070
	FNTRY CLSCOV(COM,DEV,MEAN,VAR,PTS,GO,	FLD01080
	*TITLE,NOFEAT,MAXFET,VARSIZ)	FLD01090
	IF (GO.EQ.0) GO TO 20	FLD01100
	WRITE (A,HEAD)	FLD01110
	WRITE (A,10) NOFEAT	FLD01120
	WRITE (A,100) TITLE (DASH,I=1,5)	FLD01130
	160 FORMAT(147,'TRAINING SUBCLASS' ,A4/T4H,4A4,A2)	FLD01140
	GO TO 20	FLD01150
	END	FLD01160
		FLD01170
		FLD01180
		FLD01190
		FLD01200
		FLD01210

FILE LEARN

```

SUBROUTINE LEARN(SPEC,COVAR,AVAR,CLSDES,SUBSAV,
*          FLDMEN,FLDVAK,SUBMEN,SUBVAK,SUBSTD,
*          SUBCLS,HFTALY,HSTALY,FLUSAV,SAVERT,KEPPTS,MAXSUB)
C
  IMPLICIT INTEGER(A-Z)
  REAL XSCALE,XSHFT,
  * COVAR(VARSIZ),AVAR(NUFEAT,MAXSUB),SUBSTD(NUFEAT,MAXSUB)
  DOUBLE PRECISION FLDMEN(NUFEAT),FLDVAK(VARSIZ),SUBMEN(NUFEAT),
  * SUBVAK(VARSIZ),COR(465),DEV(30)
C
  INCLUDE CUM6K4,LIST
  INCLUDE CUM6K8,LIST
  INCLUDE CUM6K6,LIST
  DIMENSION HED1(15),HED2(15),DATE(3),CUMEN(15)
  EQUIVALENCE (HED1(1),HEAD(4)),(DATE(1),HEAD(22)),
  * (HED2(1),HEAD(30)),(CUMEN(1),HEAD(48))
  2 COMMON/GLOBAL/ EAD(63),MAPTAP,DATE,SAVTAP,BMFILE,BMKEY,
  * HFILE,HISKEY,TRFUM,ERLPT,EXPKEY,MAPUNT,NUFILE,
  * DRUMAD,DRUMAS,PASIZ,DATFIL,STAFIL,ASAV,ASAVFL
  * ,NHSTUN,NHSTFI,ICIRUN,MAPFIL
  * ,DUTUNT,DUTFIL,NCHPAS,TRNSFL,BMTKFL,HISTFL,PCHUNT,
  * CRDUNT,PKTUNT,KANDIU
C STAT COMMON BLOCK
  COMMON /STBASE/SUBSV1,SUBMN1,SUBVR1,SUBSD1,SUBCL1,SAVER1,HSTAL1,
  * SPEC1,CUVAK1,AVAK1,CLSID1,FLMEN1,FLVAK1,HFTAL1,FLUSV1
  COMMON /STCBLK/ MAXFET,MAXCLS,MAXFLD,NUFEAT,NUFET2,
  * VARSIZ,NUSPEC,NUMIST,SPCBAS,IBLOCK(30),FEIVEC(30),
  * FETVC2(30),HISVEC(30),NUFLD,NUCLS,
  * FLDINF(6),FLDPTS,CLSPTS,XSIZ,XHSH,XLUW,YSIZ
C$END
C
  DIMENSION LHIST(30),VERTCS(2,11),FL(8)
  DIMENSION SPEC(5,NUSPEC),FLDSAV(10,MAXFLD),CLSDES(MAXSUB),
  * HFTALY(NUMIST,XSIZ),HSTALY(NUMIST,XSIZ),KEPPTS(MAXSUB),
  * IDATA(12000),DUMPTR(5),SUBSAV(4,MAXSUB),
  * SAVERT(22,MAXFLD),SUBCLS(1),SUBDES(500)
C
  EQUIVALENCE (LHIST,BESTVC)
  EQUIVALENCE (IBLOCK(1),NUTHG),(IBLOCK(2),PCHKEY),
  * (IBLOCK(3),SSFKEY),(IBLOCK(4),CFDKEY),
  * (IBLOCK(5),HSBKEY),(IBLOCK(6),HFDKEY),
  * (IBLOCK(7),SSLKEY),(IBLOCK(8),SFDKEY),
  * (IBLOCK(9),NUTHG2),(IBLOCK(10),CALKEY),
  * (IBLOCK(11),PCFKEY),(IBLOCK(12),PCCLKY),
  * (IBLOCK(13),TSTKEY),(IBLOCK(14),TRNKEY),
  * (IBLOCK(15),THRSKY),(IBLOCK(16),STATKY),
  * (IBLOCK(17),PCALKY)
  EQUIVALENCE (IDATA(1),COR(1))
  EQUIVALENCE (FLDINF(1),LINSTR),(FLDINF(2),LINEND),
  * (FLDINF(3),LININC),(FLDINF(4),SAMSTR),
  * (FLDINF(5),SAMEND),(FLDINF(6),SAMINC),
  * (FLDINF(7),FLDIYP)
C
  DATA ENDCKD/'$END'/,DUMPTR/1,3*0,1/,DASH/'----'/,
  * BLANK/' '/,OPAK/' ('/,CPAR/' '/,CUMMA/' ',PUNCH/7/
C
  -----
  INIZ
  -----
C*
  SET UP LOGICAL ARRAY FOR FEATURES TO BE HISTOGRAMMED.
  SUBNU=0
  DO 7 I=1,NUMIST
  DO 5 J=1,NUFEAT
  IF(HISVEC(I).EQ.FETVEC(J))GO TO 6
  5 CONTINUE
  6 LHIST(I)=J
  7 CONTINUE
  DO 922 I=1,MAXSUB
  922 SUBCLS(I)=0
C*
  READ HEADER RECORD ON DATA TAPE
  CALL TAPHDR(DATE,DATFIL)
  REWIND SAVTAP
  IF(STAFIL.EQ.0) GO TO 541
  CALL FSDSFL(SAVTAP,STAFIL,NSTAT)
  IF(NSTAT.EQ.0) GO TO 541
  WRITE(6,542) NSTAT

```

FILE LEARN

```

542 FORMAT(' BAD POSITIONING OF SAVTAP, TERMINATING ',I3)      LEA00800
CALL CMEKK                                                    LEA00810
541 CONTINUE                                                    LEA00820
CALSW = CALKEY                                                 LEA00830
BADFLG = 0                                                      LEA00840
NOFLD = 0                                                        LEA00850
NOCLS = 0                                                        LEA00860
10 CONTINUE                                                    LEA00870
WRITE(6,HEAD)                                                  LEA00880
IF(HSBKEY+HFDKEY.EQ.0) GO TO 14                                LEA00890
XSCALE=FLOAT(1-XSIZ)/(XHGH-XLOW)                               LEA00900
XSHFT=-XHGH*XSCALE+1.0                                         LEA00910
DO 20 I=1,XSIZ                                                  LEA00920
DO 20 J=1,NUHIST                                                LEA00930
20 HFTALY(J,I) = 0                                             LEA00940
IF(HSBKEY+HFDKEY.NE.2) GO TO 14                                LEA00950
DO 30 I=1,XSIZ                                                  LEA00960
DO 30 J=1,NUHIST                                                LEA00970
30 HSTALY(J,I)=0                                                LEA00980
GO TO 14                                                        LEA00990
C CLASSES                                                       LEA01000
11 READ(30,12) TITLE                                           LEA01010
12 FORMAT(10X,A4)                                              LEA01020
REWIND 30                                                       LEA01030
NOCLS=NOCLS+1                                                  LEA01040
CLSTOT=NOCLS                                                    LEA01050
CLSDS(NUCLS)=TITLE                                             LEA01060
GO TO 14                                                        LEA01070
C SUBCLASSES                                                    LEA01080
13 READ(30,12) TITLE                                           LEA01090
REWIND 30                                                       LEA01100
SUBNU=SUBNU+1                                                  LEA01110
IF(SUBNU.GT.MAXSUB) GO TO 490                                  LEA01120
SCLTOT=SUBNU                                                    LEA01130
SUBSAV(4,SUBNU)=TITLE                                          LEA01140
C STARTING FIELD                                                LEA01150
SUBSAV(2,SUBNU) = NOFLD + 1                                     LEA01160
SUBSAV(1,SUBNU)=NOCLS                                          LEA01170
SUBCLS(NUCLS)=SUBCLS(NUCLS)+1                                  LEA01180
C READ FIELD CARD                                              LEA01190
14 CFLAG=LAREAD(FLDNAM,VERTCS,FLDINF,NC)                      LEA01200
C END, CLASS, AND SUBCLASS                                     LEA01210
IF (CFLAG .EQ. 0) GO TO 60                                     LEA01220
IF (CFLAG .EQ. -1) GO TO 11                                    LEA01230
IF (CFLAG .EQ. -2) GO TO 13                                    LEA01240
NOFLD=NOFLD+1                                                  LEA01250
IF(NOFLD.GT.MAXFLD) GO TO 510                                  LEA01260
FLDSAV(1,NOFLD)=FLDNAM                                         LEA01270
FLDSAV(2,NOFLD)=NOCLS                                          LEA01280
FLDSAV(3,NOFLD)=SUBNU                                          LEA01290
FLDSAV(4,NOFLD)=NC                                             LEA01300
FLDSAV(5,NOFLD)=LINSTR                                         LEA01310
FLDSAV(6,NOFLD)=LINEND                                         LEA01320
FLDSAV(7,NOFLD)=SAMSTR                                         LEA01330
FLDSAV(8,NOFLD)=SAMEND                                         LEA01340
FLDSAV(9,NOFLD)=LININC                                         LEA01350
FLDSAV(10,NOFLD)=SAMINC                                        LEA01360
SUBSAV(3,SUBNU) = NOFLD                                        LEA01370
K=0                                                            LEA01380
DO 111 J=1,11                                                  LEA01390
DO 111 I=1,2                                                    LEA01400
K=K+1                                                           LEA01410
111 SAVERT(K,NOFLD)=VERTCS(I,J)                                LEA01420
TOTVRT=TOTVRT+NC                                              LEA01430
GO TO 14                                                        LEA01440
60 CONTINUE                                                    LEA01450
WRITE(SAVTAP) NOCLS,SUBNU,NOFEAT,NOFLD,TOTVRT,(FETVEC(I),I=1,NOFLD) LEA01460
* T)                                                           LEA01470
DO 61 I=1,NOFLD                                                LEA01480
TNC=2+FLDSAV(4,I)                                             LEA01490
WRITE(SAVTAP) (FLDSAV(J,I),J=1,4)                             LEA01500
61 WRITE(SAVTAP) (SAVERT(J,I),J=1,TNC)                        LEA01510
WRITE(SAVTAP) (CLSDS(J),J=1,NUCLS), (SUBCLS(J),J=1,NOCLS),   LEA01520
* (SUBSAV(4,J),J=1,SUBNU)                                       LEA01530
IF(PCHKEY.NE.1) GO TO 62                                       LEA01540
WRITE(PCHUNT,63)                                              LEA01550
63 FORMAT('MODULE TRAINING FIELD DECK')                       LEA01560
WRITE(PCHUNT,64) NOCLS,SUBNU,NOFEAT,NOFLD,TOTVRT            LEA01570
64 FORMAT('NUCLS ',I4,' NUSUB ',I2,' NOFEAT ',I2,' NOFLD ',I3, LEA01580

```


FILE LEARN

```

* ' TUTVRT ',14)
WRITE(PCHUNT,165) (FETVEC(I),I=1,NUFEAT)
165 FORMAT('CHNVEC',4X,30I2)
DO 65 I=1,NUFLD
WRITE(PCHUNT,66) (FLDSAV(J,I),J=1,4)
66 FORMAT(A4,6X,12,8X,12,8X,12)
TNC=2*FLDSAV(4,1)
65 WRITE(PCHUNT,67) (SAVERT(J,I),J=1,TNC)
67 FORMAT('VERTICES ',14I5)
WRITE(PCHUNT,68) (CLSDES(J),J=1,NUCLS)
68 FORMAT('CLSDES ',9(2X,A4))
WRITE(PCHUNT,69) (SUBCLS(J),J=1,NUCLS)
69 FORMAT('SUBNU ',24(1X,I2))
WRITE(PCHUNT,90) (SUBSAV(4,J),J=1,SUBNU)
90 FORMAT('SUBDES ',10(A4,1X))
62 CONTINUE
WRITE(6,41) (DASH,I=1,20)
DO 40 K=1,NUFLD
JJ=2*(FLDSAV(4,K)-1)
MP=FLDSAV(2,K)
KJ=10
IF(JJ.LE.10) KJ=JJ
MPP=FLDSAV(3,K)
WRITE(6,42) K,FLDSAV(1,K),CLSDES(MP),SUBSAV(4,MPP),FLDSAV(10,K),
*FLDSAV(9,K),((OPAK,SAVERT(I,K),CUMMA,SAVERT(I+1,K),CPAK),I=1,KJ,2)
IF(JJ.LE.10) GO TO 2017
WRITE(6,43) ((OPAK,SAVERT(I,K),CUMMA,SAVERT(I+1,K),CPAK),I=1,JJ,2)
*)
2017 CONTINUE
42 FORMAT(4X,I4,2X,A4,4X,A4,4X,A4,5X,I4,3X,I4,4X,
*5(A1,I4,A1,I4,A1,2X))
43 FORMAT(50X,5(A1,I4,A1,I4,A1,2X))
40 CONTINUE
41 FORMAT(/T50,'TRAINING FIELDS'/T49,4A4//
*7X,'FIELD',I36,'SAMPLE LINE'//
*5X,'NU. NAME',4X,'CLASS',3X,'SUBCLASS INC INC',
*4X,'VERTICES(SAMPLE,LINE)'/
*4X,3A4,2X,A4,A2,2X,2A4,1X,3A4,4X,5A4,A1)
SUBNU=0
70 SUBNU=SUBNU+1
SUBPTS=0
DO 71 I=1,NUFEAT
71 SUBMEN(I)=0.
DO 72 I=1,VARSI2
72 SUBVAR(I)=0.
FIELD1=SUBSAV(2,SUBNU)
FIELDL=SUBSAV(3,SUBNU)
DO 301 N=FIELD1,FIELDL
DO 73 I=1,NUFEAT
73 FLDMEN(I)=0.
DO 74 I=1,VARSI2
74 FLDVAR(I)=0.
LINSTR=FLDSAV(5,N)
LINEND=FLDSAV(6,N)
LININC=FLDSAV(9,N)
SAMSTR=FLDSAV(7,N)
SAMEND=FLDSAV(8,N)
SAMINC=FLDSAV(10,N)
CALL FLDINT(FLDINF,FETVEC,NUFEAT)
LINES=(LINEND-LINSTR)/LININC+1
PTS=(SAMEND-SAMSTR)/SAMINC+1
NSAMP=PTS
FLDPTS=0
DO 17 JLINES=1,LINES
CALL LINERD(IDATA,ENDTAP)
IF(ENDTAP.EQ.-1) GO TO 16
IF(JLINES.NE.1) GO TO 8
NLINES=LINSTR
GO TO 9
8 NLINES=NLINES+LININC
9 CONTINUE
K=0
DO 93 I=1,11
DO 93 J=1,2
K=K+1
93 VERTCS(J,I)=SAVERT(K,N)
CALL FLDINT(VERTCS,FLDSAV(4,N),FL,NLINES,NS,JJJ)
KK=0
NN=1

```

LEA01590
 LEA01600
 LEA01610
 LEA01620
 LEA01630
 LEA01640
 LEA01650
 LEA01660
 LEA01670
 LEA01680
 LEA01690
 LEA01700
 LEA01710
 LEA01720
 LEA01730
 LEA01740
 LEA01750
 LEA01760
 LEA01770
 LEA01780
 LEA01790
 LEA01800
 LEA01810
 LEA01820
 LEA01830
 LEA01840
 LEA01850
 LEA01860
 LEA01870
 LEA01880
 LEA01890
 LEA01900
 LEA01910
 LEA01920
 LEA01930
 LEA01940
 LEA01950
 LEA01960
 LEA01970
 LEA01980
 LEA01990
 LEA02000
 LEA02010
 LEA02020
 LEA02030
 LEA02040
 LEA02050
 LEA02060
 LEA02070
 LEA02080
 LEA02090
 LEA02100
 LEA02110
 LEA02120
 LEA02130
 LEA02140
 LEA02150
 LEA02160
 LEA02170
 LEA02180
 LEA02190
 LEA02200
 LEA02210
 LEA02220
 LEA02230
 LEA02240
 LEA02250
 LEA02260
 LEA02270
 LEA02280
 LEA02290
 LEA02300
 LEA02310
 LEA02320
 LEA02330
 LEA02340
 LEA02350
 LEA02360
 LEA02370

FILE LEARN

```

KC=0
DO 1 J=1,NUFEAT
INDEX1=(J-1)*NSAMP
JJ=J
DO 3 K=1,JJ
KK=KK+1
INDEX2=(K-1)*NSAMP
L=1
DO 2 JPT=1,NSAMP
KPT=(JPT-1)*SAMINC+SAMSTR
DO 4 JK=L,JJ,2
IF(KPT.LT.FL(JK)) GO TO 2
IF(KPT.GT.FL(JK+1)) GO TO 15
IDJ=IDATA(INDEX1+JPT)
IF(J.EQ.1) FLUPTS=FLUPTS+1
IF(K.EQ.1) FLUMEN(J)=FLUMEN(J)+IDJ
FLUVAR(KK)=FLUVAR(KK)+IDJ*IDATA(INDEX2+JPT)
IF(LHIST(NN).NE.J) GO TO 2
KC=1
IPUT=IDATA(JPT+INDEX1)*XSCALE+XSHFT+0.501
IF(IPUT.LT.1) IPUT=1
IF(IPUT.GT.XSIZ) IPUT=XSIZ
HFTALY(NN,IPUT)=HFTALY(NN,IPUT)+1
IF(JPT.EQ.NSAMP) NN=NN+1
GO TO 2
15 L=L+2
IF(L.GT.JJJ) GO TO 53
4 CONTINUE
2 CONTINUE
IF(JPT.EQ.NSAMP) GO TO 3
53 IF(KC.EQ.1) NN=NN+1
KC=0
3 CONTINUE
1 CONTINUE
17 CONTINUE
16 CONTINUE
SUBPTS=SUBPTS+FLUPTS
DO 200 I=1,NUFEAT
200 SUBMEN(I)=SUBMEN(I)+FLDMEN(I)
DO 21 I=1,VARsiz
21 SUBVAR(I)=SUBVAR(I)+FLDVAR(I)
TITLE=SUBSAV(4,SUBNO)
IF(CFKEY+SFOKEY+HFDKEY.EQ.0) GO TO 301
IF(CFKEY+SFOKEY.EQ.0) GO TO 280
CALL FLDCOV(COV,DEV,FLDMEN,FLUVAR,FLUPTS,CFKEY,FLDSAV(1,N),
*NOFEAT,MAXFET,VARsiz)
C PLOT SPECTRAL RESPONSE FOR FIELDS
IF(SFOKEY.EQ.0) GO TO 270
CALL FLDSPC(FLUMEN,DEV,FLDSAV(1,N),DUMPTK,(DATA,
*FLUMEN,FLDVAR,TITLE,NUFEAT,FETVEC,SPCBAS)
270 IF(HFDKEY.EQ.0) GO TO 301
280 IF(HSBKEY.EQ.0) GO TO 300
DO 290 I=1,XSIZ
DO 290 J=1,NOHIST
290 HSTALY(J,I)=HSTALY(J,I)+HFTALY(J,I)
300 CONTINUE
CALL FLDHIS(HFTALY,IDATA,FLDSAV(1,N),XSIZ,XMGH,XLOW,YSIZ,
*NOHIST,FLDPTS,TITLE,HISVEC)
301 CONTINUE
C CALCULATE COVAR MTX AND MEAN VECTOR FOR SUBCLASS
CALL CLSCOV(COV,DEV,SUBMEN,SUBVAR,SUBPTS,SSFKEY,
*TITLE,NUFEAT,MAXFET,VARsiz)
C SAVE SUBCLASS MEAN,COVAR,STD DEV
130 DO 31 I=1,NUFEAT
AVAR(I,SUBNO)=SUBMEN(I)
31 SUBSTD(I,SUBNO)=DEV(I)
DO 32 J=1,VARsiz
32 COVAR(J)=SUBVAR(J)
KEPPTS(SUBNO)=SUBPTS
C PLOT SPECTRAL RESPONSE FOR EACH SUBCLASS
IF(SSFKEY.EQ.0) GO TO 33
CALL CLSSPC(AVAR(1,SUBNO),SUBSTD(1,SUBNO),TITLE,DUMPTK,IDATA,
*TITLE,NUFEAT,FETVEC,SPCBAS)
C PRINT SUBCLASS HIST
33 IF(HSBKEY.EQ.0) GO TO 390
TITLE=SUBSAV(4,SUBNO)
IF(HFDKEY.EQ.0) GO TO 380
CALL CLSHIS(HSTALY,IDATA,TITLE,XSIZ,XMGH,XLOW,YSIZ,
*NOHIST,FLDPTS,HISVEC)

```

LEAU2380
LEAU2390
LEAU2400
LEAU2410
LEAU2420
LEAU2430
LEAU2440
LEAU2450
LEAU2460
LEAU2470
LEAU2480
LEAU2490
LEAU2500
LEAU2510
LEAU2520
LEAU2530
LEAU2540
LEAU2550
LEAU2560
LEAU2570
LEAU2580
LEAU2590
LEAU2600
LEAU2610
LEAU2620
LEAU2630
LEAU2640
LEAU2650
LEAU2660
LEAU2670
LEAU2680
LEAU2690
LEAU2700
LEAU2710
LEAU2720
LEAU2730
LEAU2740
LEAU2750
LEAU2760
LEAU2770
LEAU2780
LEAU2790
LEAU2800
LEAU2810
LEAU2820
LEAU2830
LEAU2840
LEAU2850
LEAU2860
LEAU2870
LEAU2880
LEAU2890
LEAU2900
LEAU2910
LEAU2920
LEAU2930
LEAU2940
LEAU2950
LEAU2960
LEAU2970
LEAU2980
LEAU2990
LEAU3000
LEAU3010
LEAU3020
LEAU3030
LEAU3040
LEAU3050
LEAU3060
LEAU3070
LEAU3080
LEAU3090
LEAU3100
LEAU3110
LEAU3120
LEAU3130
LEAU3140
LEAU3150
LEAU3160

FILE LEARN

```

GO TO 390
380 CALL CLSHIS(HFTALY, IDATA, TITLE ,XSIZ,XMH,XLOW,YSIZ,
*NOHIST,FLDPTS,HIS,VEC)
390 WRITE(SAVTAP) KEPPTS(SUBNO),(CUVAR(I),I=1,VAKSIZ),
* (AVAR(I,SUBNO),I=1,NUFEAT)
IF(PCMKY.NE.1) GO TO 94
WRITE(PCHUNT,95) KEPPTS(SUBNO)
95 FORMAT('NUPTS ',6X,18)
WRITE(PCHUNT,96) (AVAR(I,SUBNO),I=1,NUFEAT)
96 FORMAT('MEANS',5E15.8)
WRITE(PCHUNT,97) (CUVAR(I),I=1,VAKSIZ)
97 FORMAT('CUVAR',5E15.8)
94 IF(SUBNO.LT.SCLTOT) GO TO 70
ENDFILE SAVTAP
REWIND SAVTAP

PUBLISH THE MULTISPECTRAL PLOTS
-----
410 IF (SPEC(1,1).NE.0) GO TO 450
JK = 0
DO 430 I=1,NOSPEC
DO 420 J=1,4
II=1
JJ=J
JK = JK+1
SPEC(J,1) = JK
IF(JK.EQ.SUBNO) GO TO 440
420 CONTINUE
430 SPEC(5,I) = 4
440 SPEC(5,II)=JJ
NOSPEC=II
450 CONTINUE
DO 480 I=1,NOSPEC
K = SPEC(5,I)
JJ = 0
DO 460 J=1,K
IF(SPEC(J,I).GT.SUBNO) GO TO 460
JJ = JJ + 1
SPEC(JJ,1) = SPEC(J,I)
460 CONTINUE
IF (JJ.EQ.0) GO TO 480
SPEC(5,I) = JJ
WRITE(6,HEAD)
WRITE(6,465)
465 FORMAT(/ 27X,'COMPOSITE SPECTRAL PLOT (MEAN,PLUS AND MINUS ONE S
10. DEV. ) FOR ' / )
WRITE(6,470) (SPEC(J,I),J=1,JJ)
470 FORMAT(38X,'TRAINING SUBCLASS(ES) ',414/41X,'-----'/ )
DO 98 JI=1,SUBNO
SUBDES(JI)=SUBSAV(4,JI)
IF(JJ.LT.4) WRITE(6,471) ( DASH,II=1,5 )
471 FORMAT(41X,4A4,A2 / )
CALL MULSPC(AVAR(1,1),SUBSTD(1,1),SUBDES,SPEC(1,1),IDATA,
*NUFEAT,FETVEC,SPCBAS)
480 CONTINUE
CALL SETMKG(66,4,62)
RETURN

ERROR ROUTINES
-----
490 BADFLG = 2
WRITE(6,500) MAXSUB,MAXSUB
500 FORMAT(/5X,'*** STAT/LEARNN MAX NO. OF',18,3X, 'SUBCLASSES EXCEL
*EDED--FIRST',18,3X,'SUBCLASSES USED--REMAINDER IGNORED'///)
GO TO 530

510 BADFLG = 1
WRITE(6,520) MAXFLD,MAXFLD,SUBNO
520 FORMAT(/5X,'***** STAT/LEARNN --- MAX. OF',18,3X,
1 'FIELDS EXCEEDED --- ',18,3X,'FIELDS RETAINED FOR',18,3X,
* 'SUBCLASSES'/5X,'***REMAINDER OF INPUT TRAINING FIELDS NOT USED'/ )

530 READ (21,540)I
540 FORMAT (A4)
IF (I.NE.ENDCRD) GO TO 530
GO TO 60

```

FILE LEARN

END

LEA03960

~~8-13~~

64

FILE: SETUP1

C		SET00800
C	READ AND ANALYZE SUPERVISOR CONTROL CARDS	SET00810
C	-----	SET00820
C	SETUP REREAD BUFFER	SET00830
C	CALL REREAD(30,80)	SET00840
C	200 COL = 0	SET00850
C	NOW READ THE CARD INTO THE BUFFER	SET00860
C	READ(21,150)(ACARD(I),I=1,20)	SET00870
C	150 FORMAT(20A4)	SET00880
C	WRITE(30,150)(ACARD(I),I=1,20)	SET00890
C	REWIND 30	SET00900
C	2002 READ(30,2002) CODE,CARD2	SET00910
C	FORMAT (A4,6X,62A1)	SET00920
C	REWIND 30	SET00930
C	225 WRITE(6,2252) CODE,CARD2	SET00940
C	2252 FORMAT (15,A4,6X,62A1)	SET00950
C	DO 230 I=1,CINMAX	SET00960
C	IF (CINDEX(I) .EQ. CODE)	SET00970
C	1 GO TO(10,600,700,800,1100,1200,1400,900,1500,1600,	SET00980
C	2 1700,1710,1720),I	SET00990
C	230 CONTINUE	SET01000
C	GO TO 1000	SET01010
C	OPTION CARD	SET01020
C	-----	SET01030
C	10 M = NXTCHR(CARD2,COL)	SET01040
C	IF (M .EQ. BLANK) GO TO 200	SET01050
C	IF (M .EQ. OPTCOD(1)) GO TO 20	SET01060
C	SETFLG = 1	SET01070
C	IF (M .NE. NB CD) GOTO 14	SET01080
C	J = COL-1	SET01090
C	M = NXTCHR(CARD2,COL)	SET01100
C	IF (M .NE. OB CD) GOTO 12	SET01110
C	SETFLG = 0	SET01120
C	J = COL	SET01130
C	12 COL = J	SET01140
C	M = NXTCHR(CARD2,COL)	SET01150
C	14 DO 15 I=2,5	SET01160
C	IF (M .EQ. OPTCOD(I)) GO TO (40,30,25,30,30), I	SET01170
C	15 CONTINUE	SET01180
C	IF (M .EQ. BLANK) GO TO 200	SET01190
C	40 M = COL + 10	SET01200
C	WRITE(6,402) M	SET01210
C	402 FORMAT(/ 1X,1** STAT/SETUP1 --- ERROR IN OPTION(S) REQUESTED - S	SET01220
C	1CAN OF OPTION(S) DISCONTINUED AT CARD COLUMN',15,2X,1**' /)	SET01230
C	GO TO 200	SET01240
C	20 M = FIND12(CARD2,COL,SINVEC)	SET01250
C	IF (SINVEC(M) .NE. EQUAL) GO TO 40	SET01260
C	M = NUMBER(CARD2,COL,NUMVEC,29)	SET01270
C	IF (NUMVEC(30) .LE. 0) GO TO 40	SET01280
C	MAXSUB=NUMVEC(30)	SET01290
C	GO TO 10	SET01300
C	25 J = 20	SET01310
C	M = NXTCHR(CARD2,COL)	SET01320
C	IF (M .EQ. OB CD) J=3	SET01330
C	IF (M .EQ. AB CD) J=9	SET01340
C	IF (J .LT. 20) GOTO 32	SET01350
C	GOTO 40	SET01360
C	30 J = 1*2-3	SET01370
C	32 M = FIND12(CARD2,COL,SINVEC)	SET01380
C	IF (SINVEC(M) .NE. EQUAL) GOTO 38	SET01390
C	M = NXTCHR(CARD2,COL)	SET01400
C	IF (M .EQ. CB CD) IBLOCK(J) = SETFLG	SET01410
C	IF (M .EQ. FB CD) IBLOCK(J+1) = SETFLG	SET01420
C	M = FIND12(CARD2,COL,SINVEC)	SET01430
C	IF (M .LE. 0) GOTO 200	SET01440
C	GOTO 10	SET01450
C	38 IBLOCK(J) = SETFLG	SET01460
C		SET01470
C		SET01480
C		SET01490
C		SET01500
C		SET01510
C		SET01520
C		SET01530
C		SET01540
C		SET01550
C		SET01560
C		SET01570
C		SET01580

SEF0101590
SEF0101600
SEF0101610
SEF0101620
SEF0101630
SEF0101640
SEF0101650
SEF0101660
SEF0101670
SEF0101680
SEF0101690
SEF0101700
SEF0101710
SEF0101720
SEF0101730
SEF0101740
SEF0101750
SEF0101760
SEF0101770
SEF0101780
SEF0101790
SEF0101800
SEF0101810
SEF0101820
SEF0101830
SEF0101840
SEF0101850
SEF0101860
SEF0101870
SEF0101880
SEF0101890
SEF0101900
SEF0101910
SEF0101920
SEF0101930
SEF0101940
SEF0101950
SEF0101960
SEF0101970
SEF0101980
SEF0101990
SEF0202000
SEF0202010
SEF0202020
SEF0202030
SEF0202040
SEF0202050
SEF0202060
SEF0202070
SEF0202080
SEF0202090
SEF0202100
SEF0202110
SEF0202120
SEF0202130
SEF0202140
SEF0202150
SEF0202160
SEF0202170
SEF0202180
SEF0202190
SEF0202200
SEF0202210
SEF0202220
SEF0202230
SEF0202240
SEF0202250
SEF0202260
SEF0202270
SEF0202280
SEF0202290
SEF0202300
SEF0202310
SEF0202320
SEF0202330
SEF0202340
SEF0202350
SEF0202360
SEF0202370

```

DO 714 I=1,NM1,1
IP1 = I + 1
IF( IP1 .GT. NOHIST) GO TO 714
DO 713 J=IP1,NOHIST,1
IF( HISVEC(I).LT. HISVEC(J)) GO TO 713
TEMP = HISVEC(I)
HISVEC(I) = HISVEC(J)
HISVEC(J) = TEMP
713 CONTINUE
714 CONTINUE
GO TO 200

```

```

C 800 J = NXTCHR(CARD2,COL)
      IF ( J.EQ. BLANK ) GOTO 200
      COL = COL+1
      NOSPEC = NOSPEC + 1
      IF( NOSPEC.GT. 20) GO TO 200
      J = NUMBER(CARD2,COL,NUMVEC,0)
      IF ( J.GT. 4 ) J = 4
      DO 810 I=1,J
      IF (NUMVEC(I) .LE. 0 ) GOTO 815
810  SPVEC(I,NOSPEC) = NUMVEC(I)
      I = J+1
815  SPVEC(5,NOSPEC) = I-1
      GOTO 200

```

```

1100 J = NXTCHR(CARD2,COL)
      IF ( J .EQ. BLANK ) GOTO 200
      COL = COL+1
      NBLOCK = NUMBER(CARD2,COL,NUMVEC,NBLOCK)
      DO 1110 I=1,NBLOCK,1
        IF (NUMVEC(I) .EQ. 1) IBLOCK(I) = 1
1110 CONTINUE
      GO TO 200

```

```

C      97 COL=COL-1
1200  J= NXTCHK (CARD2,COL)
      IF ( J.EQ. BLANK ) GO TO 200
      IF ( J.EQ. XHCD ) GO TO 1220
      IF ( J.EQ. SHCD ) GO TO 1230
      IF ( J.EQ. YBCD ) GO TO 1240
      GO TO 1000

```

```

C      60 TO 1000
1220 J = NXTCHN(CARD2,COL)
      M = FIND12(CARD2,COL,SINVEC)
      IF( SINVEC(M) .NE. EQUAL ) GO TO 1000
      M = NUMBER(CARD2,COL,NUMVEC,29)
      IF( J .EQ. LHCD ) XLOW = NUMVEC(30)
      IF( J .EQ. HBKD ) XHIGH = NUMVEC(30)
      IF( J .NE. SRCD ) GO TO 97
      XSIZ = NUMVEC(30)
      GO TO 97

```

```

C 1230 M = FIND12(CARD2,COL,SINVEC)
      IF( SINVEC(M) .NE. EQUAL ) GO TO 1000
      M = NUMBER( CARD2,COL,NUMVEC,29)
      SPCBAS = NUMVEC(30)
      GO TO 97

```

```

C 1240 M = FIND12(CARD2,COL,SINVEC)
      IF( SINVEC(M) .NE. EQUAL ) GO TO 1000
      M = NUMBER(CARD2,COL,NUMVEC,29)
      YSIZ = NUMVEC(30)
      GO TO 97

```

```
1400 M = NIXTCHR(CARD2,COL)
      IF( M .EQ. BLANK ) GO TO 200
```

SET023A0
SET02390
SET02400
SET02410
SET02420
SET02430
SET02440
SET02450
SET02460
SET02470
SET02480
SET02490
SET02500
SET02510
SET02520
SET02530
SET02540
SET02550
SET02560
SET02570
SET02580
SET02590
SET02600
SET02610
SET02620
SET02630
SET02640
SET02650
SET02660
SET02670
SET02680
SET02690
SET02700
SET02710
SET02720
SET02730
SET02740
SET02750
SET02760
SET02770
SET02780
SET02790
SET02800
SET02810
SET02820
SET02830
SET02840
SET02850
SET02860
SET02870
SET02880
SET02890
SET02900
SET02910
SET02920
SET02930
SET02940
SET02950
SET02960
SET02970
SET02980
SET02990
SET03000
SET03010
SET03020
SET03030
SET03040
SET03050
SET03060
SET03070
SET03080
SET03090
SET03100
SET03110
SET03120
SET03130
SET03140
SET03150
SET03160

FILE: SETUP1

```

      READ (30,999) DATE
999  FORMAT (10X,15A4)
      REWIND 30
      GO TO 200

      COMMENT CARD
      -----

1500 READ (30,999) COMENT
      REWIND 30
      GO TO 200

      HED1 CARD
      -----

1600 READ (30,999) HED1
      REWIND 30
      GO TO 200

      HED2 CARD
      -----

1700 READ (30,999) HED2
      REWIND 30
      GOTO 200

      DATA FILE CARD

1710 M = NXTCHR(CARD2,COL)
      IF (M.EQ. BLANK) GO TO 200
      IF (M.EQ. UNCD) GO TO 1715
      IF (M.EQ. FBCD) GO TO 1717
1713 WRITE(6,753)
753  FORMAT(' ERROR ON DATA FILE CARD')
      GO TO 200
1715 J = FIND12(CARD2,COL,EQUVEC)
      IF (J.EQ. -1) GO TO 1713
      M = NUMBER(CARD2,COL,DATAPE,ZERO)
      COL = COL - 1
      GO TO 1710
1717 J = FIND12(CARD2,COL,EQUVEC)
      IF (J.EQ. -1) GO TO 1713
      M = NUMBER(CARD2,COL,DATFIL,ZERO)
      DATFIL = DATFIL - 1
      IF (DATFIL.LT. 0) DATFIL = 0
      COL = COL - 1
      GO TO 1710

      STAT FILE CARD

1720 M = NXTCHR(CARD2,COL)
      IF (M.EQ. BLANK) GO TO 200
      IF (M.EQ. UNCD) GO TO 1725
      IF (M.EQ. FBCD) GO TO 1727
1723 WRITE(6,755)
755  FORMAT(' ERROR ON STAT FILE CARD')
      GO TO 200
1725 J = FIND12(CARD2,COL,EQUVEC)
      IF (J.EQ. -1) GO TO 1723
      M = NUMBER(CARD2,COL,SAVTAP,ZERO)
      COL = COL - 1
      GO TO 1720
1727 J = FIND12(CARD2,COL,EQUVEC)
      IF (J.EQ. -1) GO TO 1723
      M = NUMBER(CARD2,COL,STAFIL,ZERO)
      STAFIL = STAFIL - 1
      IF (STAFIL.LT. 0) STAFIL = 0
      COL = COL - 1
      GO TO 1720

      CALCULATE BASES OF THE ARRAYS
      -----

900  CONTINUE
      IF (NOSPEC.GT. 20) NOSPEC = 20
      IF (NUSPEC.NE. 0) GO TO 950
      NOSPEC = (MAXCLS*3)/4
      SPCVEC(1,1) = 0

```

```

SET03170
SET03180
SET03190
SET03200
SET03210
SET03220
SET03230
SET03240
SET03250
SET03260
SET03270
SET03280
SET03290
SET03300
SET03310
SET03320
SET03330
SET03340
SET03350
SET03360
SET03370
SET03380
SET03390
SET03400
SET03410
SET03420
SET03430
SET03440
SET03450
SET03460
SET03470
SET03480
SET03490
SET03500
SET03510
SET03520
SET03530
SET03540
SET03550
SET03560
SET03570
SET03580
SET03590
SET03600
SET03610
SET03620
SET03630
SET03640
SET03650
SET03660
SET03670
SET03680
SET03690
SET03700
SET03710
SET03720
SET03730
SET03740
SET03750
SET03760
SET03770
SET03780
SET03790
SET03800
SET03810
SET03820
SET03830
SET03840
SET03850
SET03860
SET03870
SET03880
SET03890
SET03900
SET03910
SET03920
SET03930
SET03940
SET03950

```

FILE: SETUP1

```

950 VARSIZ = NOFEAT*(NOFEAT+1)/2
IF (XSIZ.LE.0) XSIZ=XHGH-XLOW+1
IF (XSIZ.GT.101) XSIZ=101
SPEC1=(5*NOSPEC+1)/2*2
COVAR1=(VARSIZ+1)/2*2
AVAR1=(NOFEAT*MAXSUB+1)/2*2
CLS101=(MAXSUB+1)/2*2
SUBSV1=(5*MAXSUB+1)/2*2
FLMEN1=NOFEAT*2
FLVAR1=VARSIZ*2
SUBMN1=NOFEAT*2
SURVR1=VARSIZ*2
SURSD1=(NOFEAT*MAXSUB+1)/2*2
SUBCL1=(MAXSUB+1)/2*2
HFTAL1=(XSIZ*NOHIST+1)/2*2*HFOKEY
HSTAL1=(XSIZ*NOHIST+1)/2*2*HSBKEY
SIZE=SPEC1*COVAR1*AVAR1*CLS101*SUBSV1*FLMEN1*FLVAR1*
* SUBMN1*SURVR1*SURSD1*SUBCL1*HFTAL1*HSTAL1
MAXFLD=(TOP-SIZE-32)/32
IF (MAXFLD.LE.0) GO TO 1300
SPEC1=1
COVAR1=SPEC1*(5*NOSPEC+1)/2
AVAR1=COVAR1*(VARSIZ+1)/2
CLS101=AVAR1*(NOFEAT*MAXSUB+1)/2
SUBSV1=CLS101*(MAXSUB+1)/2
FLMEN1=SUBSV1*(5*MAXSUB+1)/2
FLVAR1=FLMEN1*NOFEAT
SUBMN1=FLVAR1*VARSIZ
SURVR1=SUBMN1*NOFEAT
SURSD1=SURVR1*VARSIZ
SURCL1=SURSD1*(NOFEAT*MAXSUB+1)/2
HFTAL1=SURCL1*(MAXSUB+1)/2
HSTAL1=HFTAL1*(XSIZ*NOHIST+1)/2*HFOKEY
FLDSV1=HSTAL1*(XSIZ*NOHIST+1)/2*HSBKEY
SAVER1=FLDSV1*(10*MAXFLD)/2
TIPTOP=SAVER1*(22*MAXFLD)/2
BADCOR=TOP-2*TIPTOP
IF (BADCOR.LI.0) GO TO 1300
PRINT OUT OPTIONS
-----
C
WRITE(6,HEAD)
IF (PCHKEY+SSFKEY+CFDKEY+HSHKEY+HFOKEY+SSLKEY
1 *SFOKEY+CALKEY.LE.0) GOTO 960
WRITE(6,9001)
IF (CFDKEY.EQ.1) WRITE(6,9002)
IF (SFOKEY.EQ.1) WRITE(6,9004)
IF (SSLKEY.EQ.1) WRITE(6,9006)
IF (PCHKEY.EQ.1) WRITE(6,9008)
IF (HFOKEY.EQ.1) WRITE(6,9012)
IF (HSHKEY.EQ.1) WRITE(6,9014)
IF (CALKEY.EQ.1) WRITE(6,9016)
IF (SSFKEY.EQ.1) WRITE(6,9018)
C
9001 FORMAT(1X,'YOU HAVE SELECTED THE FOLLOWING $STAT PROCESSOR OPTIO
INS: '///)
9002 FORMAT(15,' PRINT MEAN AND COVARIANCE FOR EACH FIELD')
9004 FORMAT(15,' PRINT SPECTRAL PLOT FOR EACH FIELD')
9006 FORMAT(15,' PRINT SPECTRAL PLOT FOR EACH SUBCLASS')
9008 FORMAT(15,' PUNCH MEAN AND COVARIANCE MATRIX FOR EACH SUBCLASS')
9012 FORMAT(15,' PRINT A HISTOGRAM FOR EACH FIELD')
9014 FORMAT(15,' PRINT A HISTOGRAM FOR EACH SUBCLASS')
9016 FORMAT(15,' *** USE CALIBRATED DATA ***')
9018 FORMAT(15,' PRINT MEAN AND COVARIANCE FOR EACH SUBCLASS')
960 WRITE(6,9502) BADCOR,MAXFLD,MAXSUB,(FETVEC(I),I=1,NOFEAT)
9502 FORMAT(10,'//0SUPERVISOR INFORMATION: //15,'UNUSED CORE',16,'LOCAS
ATIONS',15,'MAXIMUM NO. OF FIELDS...',13/15,'MAXIMUM NO. OF SUBCLAS
2ES...',13/15,'CHANNELS SELECTED ARE ',15(13,' ')/127,15(13,' ')
IF (HFOKEY+HCLKEY.NE.0) WRITE(6,9504) (HISVEC(I),I=1,NOHIST)
9504 FORMAT(15,'HISTOGRAM CHANNELS ARE ',15(13,' ')/128,15(13,' ')
IF (SPCVEC(1,1).EQ.0) GOTO 971
WRITE(6,9505)
9505 FORMAT(15,'MULTISPECTRAL PLOTS ARE...')
DO 970 J=1,NOSPEC
K = SPCVEC(5,J)
WRITE(6,9506) (SPCVEC(I,J),I=1,K)
9506 FORMAT(144,' ')/T31,'(')/4(12,' '))
970 CONTINUE
971 CONTINUE

```

FILE: SETUP1

980 RETURN	SET04750
C	SET04760
ERROR ROUTINES	SET04770
-----	SET04780
C	SET04790
1000 WRITE (6,10002) CODE, CARD2	SET04800
10002 FORMAT(///5X, '///// FROM SUBR. SETUP1 --- BAD CONTROL CARD ENC	SET04810
10UNTERED --- INPUT CARD IS ...//8X,2H',A4.6X,62A1.2H', //)	SET04820
GO TO 200	SET04830
C	SET04840
1300 WRITE (6,1302)	SET04850
1302 FORMAT(///5X, '///// FROM SUBR. SETUP1 --- DECREASE OPTIONS'	SET04860
1 // 5X, '***** TERMINATING PROGRAM EXECUTION FROM SUBR. SETUP1 *	SET04870
2 *** /1H1)	SET04880
GO TO 1305	SET04890
1303 WRITE (6,10002) CODE, CARD2	SET04900
WRITE (6,13031) MAXFET	SET04910
13031 FORMAT(//5X, 'CHECK CHANNELS ON SUBCLASS NO.S REQUESTED-CANNOT BE	SET04920
1 LESS THAN OR EQUAL ZERO, OR GREATER THAN', 15//5X,	SET04930
2 ***** TERMINATING PROGRAM EXECUTION FROM SUBR. SETUP1 *****	SET04940
3 /1H1)	SET04950
1305 CALL EXIT	SET04960
END	SET04970

9. ISOCLS PROCESSOR

See listings for the TESTSP processor (section 23) for an iterative self-organizing clustering procedure using sample values of pixels clustered in packed form on disk storage.

FILE ISOCLS

```

SUBROUTINE ISOCLS (ARRAY, TOP)
C*****
C THIS PROGRAM PERFORMS A MODIFIED VERSION OF THE CLUSTERING
C ALGORITHM (ISODATA) ORIGINALLY DEVELOPED BY BALL AND HALL OF
C STANFORD RESEARCH INSTITUTE. THE ALGORITHM HAS BEEN MODIFIED
C ON THE RECOMMENDATIONS OF ED KAN (LEC).
C
C THE PROGRAM EXPECTS MULTISPECTRAL SCANNER DATA
C IN EITHER THE LAPSYS 22 OR THE UNIVERSAL
C FORMAT. THE DATA TAPE SHOULD BE ASSIGNED TO FORTRAN UNIT 3.
C*****
C IMPLICIT INTEGER (A-X)
C INCLUDE COMBK5, LIST
C INCLUDE COMNT5, LIST
C INCLUDE COMBK6, LIST
C INCLUDE COMBK16, LIST
COMMON /PASS/ STOP, LNCAT, NMIN, KRN, STOMAX, DLMIN, SEP,
* MAP, SPTRIG, IRD, KPTS, NOPTS, PUNCH,
* ICHN, CHNTHS, ICHAIN (62), NWDS, IREGIN, REGIN1,
* BEGIN2, REGIN3, CLSNAM, NOFLD, IPI, TOTWRD, TOTPTS,
* NCLASS, NOCLS, TOTSUB, TOTFLD, TOTVRT, NOCL, NVRT
* ,NXTCLS, NOFEAT, MAXCLS, FETVEC (30), SYMNTX (62)
* ,VARSI, STATKY, ISOKEY, MAPFMT, MAPKEY, SEQUEN (20), PERCEN, SIMERP
* , IORDER, INIUNIT, INFILE, INITM, PMIN, SUBVEC (62), NOSUB2, CHNVC (30)
* , NOCHAN, ERCOMP, NOSEQ, MEAND0, MEANDU,
* SYMD0, SYMDU, ITRIG0, ITRIGU, DOFLAG,
* DUFLAG, DODU, STDOTS (60), NSDOTS, SUNCOR (30), LLNCAT,
* DVERT (250, 2), DRECT (60, 2), DVPNT (11, 2), IDCNT (2), NDOU (2)
* , MXFET1, MAXPOP
REAL SUNCOR
C
C COMMON BLOCK 'PASS' IS USED ONLY BY THE ISOCLS PROCESSOR.
C
C ISOCLS USES THE RANDOM ACCESS DRUM FILE AS FOUR DISTINCT FILES.
C SEE DEFINITIONS OF IREGIN, BEGIN1, BEGIN2, BEGIN3 BELOW
C
C DEFINITIONS
C
C ISTOP - MAX. NO. OF ITERATIONS FOR THE CLUSTERING PROCEDURE
C SET IN SETUP7 ROUTINE. (USER INPUT)
C LNCAT - CURRENT NO. OF CLUSTERS. SET INITIALLY IN RDFILE OR IS
C ISOCLS. THEN ONLY IN ISODAT.
C NMIN - MIN. NO. OF POINTS TO ALLOW PER CLUSTER
C SET IN SETUP7 ROUTINE. (USER INPUT)
C KRN - PRINT CLUSTER SUMMARY EVERY 'KRN' ITERATION(S)
C SET IN SETUP7 ROUTINE. (USER INPUT)
C STOMAX - STANDARD DEVIATION FOR SPLITTING CLUSTERS
C SET IN SETUP7 ROUTINE. (USER INPUT)
C DLMIN - MIN. DISTANCE BETWEEN CLUSTERS FOR COMBINING.
C SEP - DISTANCE TO SEPARATE CLUSTERS. SET EITHER IN SETUP7,
C BY USER INPUT, OR IN ISODAT.
C MAP - PRINT A CLUSTER MAP EVERY 'MAP' ITERATION(S) - SETUP7
C SPTRIG - TRIGGER TELLING WHETHER OR NOT 'SEP' WAS INPUT. -SETUP7
C IRD - NO. OF RECORDS TO READ FROM DATA FILE. COMPUTED IN
C ISOCLS
C NOPTS - NO. OF POINTS IN EACH RECORD. COMPUTER IN ISOCLS
C CONTINUE
C KPTS - NO. OF POINTS IN LAST RECORD. COMPUTER IN ISOCLS
C PUNCH - TRIGGER TELLING WHETHER OR NOT TO PUNCH THE MODULE
C STAT DECK. - SETUP7
C ICHN - TRIGGER TELLING WHETHER OR NOT CHAINING IS TO BE DONE
C CHNTHS - MIN. DISTANCE BETWEEN CLUSTERS FOR CHAINING - SETUP7
C ICHAIN - ARRAY CONTAINING CHAINED CLUSTER NUMBERS. SET IN
C 'CHAIN' ROUTINE.
C NWDS - TOTAL NO. OF WORDS AVAILABLE FOR DRUM STORAGE OF
C IMAGE DATA TO BE CLUSTERED - SET IN ISOCLS
C IREGIN - BEGINNING DRUM FILE ADDRESS FOR INPUT INITIAL CLUSTER
C CENTERS - SET IN ISOCLS
C REGINS - BEGINNING DRUM FILE ADDRESS FOR TEMPORARY STORAGE OF
C CLASS STATISTICS - SET IN ISOCLS ROUTINE
C BEGIN1 - BEGINNING DRUM FILE ADDRESS FOR IMAGE DATA

```

FILE ISOCLS

```

C*      BEGIN2 - BEGINNING DRUM FILE ADDRESS FOR 'IPLACE' .(CLUSTER TO ISO00770
C*      WHICH CORRESPONDING POINT BELONGS.) ISO00780
C*      CLSNAM - NAME OF CLASS CURRENTLY BEING PROCESSED. - RDDATA ISO00790
C*      NOFLD - NO. OF FIELDS INPUT FOR THIS CLASS - RDDATA ISO00800
C*      IPT - NO. OF WORDS OF STORAGE USED IN 'ARRAY' FOR FIELD AND ISO00810
C*      CLASS INFORMATION FOR THIS CLASS. - RDDATA ISO00820
C*      TOTWRD - TOTAL WORDS WRITTEN ON DRUM FILE BEGINNING AT ADDRESS ISO00830
C*      BEGIN1 - RDDATA ISO00840
C*      TOTPTS - TOTAL POINTS TO BE CLUSTERED FOR CURRENT CLASS - RDDATA ISO00850
C*      NCLASS - NO. OF CLASSES TO BE CLUSTERED FOR CURRENT CALL TO ISO00860
C*      ISOCLS. USER INPUT - SETUP7. ISO00870
C*      NOCLS - CURRENT CLASS NO. - ISOCLS ISO00880
C*      TOTSUB - TOTAL CLUSTERS FOR THIS CALL TO ISOCLS ISO00890
C*      TOTFLD - TOTAL FIELDS FOR ALL CLASSES - ISOCLS ISO00900
C*      TOTVRT - TOTAL VERTICES FOR ALL FIELDS - ISOCLS ISO00910
C*      NOCL - NO. OF CLASSES SINCE LAST CALL TO SETUP - RDDATA ISO00920
C*      IEY0331 COMMENTS DELETED *****
C*      COMMON/GLOBAL/HEAD(63),MAPTAP,DATAP,SAVTAP,RMFILE,RMKEY, ISO01200
C*      HISFIL,HISKEY,TRFORM,ERIPTP,ERPKEY,MAPUNT,NOFILE, ISO01210
C*      * DRUMAD,DRMWDS,PAGSIZ,DATFIL,STAFIL,ASAV,ASAVFL ISO01220
C*      * .NHSTUN,NHSTFI,SCTRUN,MAPFIL ISO01230
C*      * .DOTUNT,DOTFIL,NCHPAS,TRNSFL,BMTRFL,HISTFL,PCHUNT, ISO01240
C*      * CRDUNT,PRUNT,RANDIO ISO01250
C*      COMMON/ISOLNK/SUNANG(8),ISUNT,ISUNC,SMSTR,SMSTP,SMINC,LINSKP ISO01260
C*      CSEND ISO01270
C*      DIMENSION KVAR(11500) ISO01280
C*      KVARDM = 11500 ISO01290
C*      DIMENSION ARRAY(1) ISO01300
C*      DIMENSION COVAR(465) ISO01310
C*      DIMENSION NN(60) ISO01320
C*      DATA SYMDA /* /*,SYMDB /* /* /* ISO01330
C*      MAXPOP=62 ISO01340
C*      MXFET1=30 ISO01350
C*      IBEGIN=DRUMAD ISO01360
C*      RESERVE ENOUGH DRUM STORAGE FOR MAXIMUM INITIAL MEANS ISO01370
C*      BEGIN3=IBEGIN + MAXPOP*MXFET1 + MXFET1 + 2 ISO01380
C*      CALL SETUP TO READ CARD INPUT AND INITIALIZE DEFAULT VALUES ISO01390
C*      ITIME=1 ISO01400
C*      NOCLS = 0 ISO01410
C*      TOTFLD = 0 ISO01420
C*      TOTVRT = 0 ISO01430
C*      TOTSUB = 0 ISO01440
C*      CORBAS=1 ISO01450
C*      ITRIGU = 0 ISO01460
C*      ITRIGO=0 ISO01470
C*      SYMDO = SYMDA ISO01480
C*      SYMDU = SYMDB ISO01490
C*      MEANDU = 0 ISO01500
C*      MEANDU = 255 ISO01510
C*      1 CALL SETUP7(ARRAY(CORBAS),TOP,ITIME) ISO01520
C*      IDUM = MAXCLS ISO01530
C*      IF(ITIME.GT.1)GO TO 2 ISO01540
C*      VARSIZ=NOFEAT*(NOFEAT+1)/2 ISO01550
C*      BEGIN1 = BEGIN3 + NCLASS*MAXPOP*(VARSIZ + NOFEAT + 1) ISO01560
C*      NWDS=DRMWDS-(BEGIN1-DRUMAD) ISO01570
C*      2 ITIME=ITIME+1 ISO01580
C*      NOCL=0 ISO01590
C*      CALL RDDATA TO COORDINATE READING OF DATA ISO01600
C*      5 MAXDIM = TOP-CORBAS ISO01610
C*      FDI=CORBAS ISO01620
C*      CALL RDDATA(ARRAY(FDI),MAXDIM,KVAR,KVARDM,LAST) ISO01630
C*      MAXCLS = IDUM + NODU ISO01640
C*      WRITE(6,210) NODU(1),NODU(2) ISO01650
C*      210 FORMAT(1X,/' DO/OU CLUSTER POP FOR THIS CLASS ',217) ISO01660
C*      BEGIN2=BEGIN1 + TOTWRD ISO01670
C*      N1 = FDI + IPT ISO01680
C*      MEANS1=N1 + MAXCLS ISO01690
C*      STDEV1=MEANS1 + MAXCLS*NOFEAT ISO01700
C*      TTOP = STDEV1 + MAXCLS*NOFEAT ISO01710
C*      MAXDIM=TOP-TTOP ISO01720

```

FILE ISOCLS

```

      NOPTS = MAXDIM/(NOFEAT+1)
      IDAT1 = TTOP
      IF (NSDOTS.EQ.0) GO TO 4
      DOTDMF = NOCHAN
      TYPSTW = 1
      CALL RDOTS (ARRAY (MEANS1), STDOTS, NSDOTS,
      * TYPSTW, DOTDMF, DOTDMC, DOTDUM, COVAR,
      * NOCHAN, CHNVC, DOTDM, COVAR,
      * DOTDM, DOTDM, DOTDM, DOTDM, DOTDM, DOTDM, KVAR)
      LNCAT = NSDOTS
      DO 500 I = 1, NSDOTS
      DO 500 K = 1, NOFEAT
      III = (I-1)*NOFEAT + K
      II = III + MEANS1 - 1
500   ARRAY (II) = KVAR (III)
      IF (NOCHAN.EQ.NOFEAT) GO TO 8
      WRITE (6,110)
110   FORMAT (1H, 'NO CHANNELS FOR STARTING NOT EQUAL THAT FOR CLUSTER')
      GO TO 9
      CONTINUE
      IF (ISOKEY.EQ.1) GO TO 7
      SURVEC = SURCLASSES FROM STATISTICS FILE FOR INTIAL MEANS.
      NOSUB2 = NUMBER OF INITIAL MEANS.
      CHNVEC = NUMBER OF CHANNELS FROM STATISTICS FILE. NOCHAN MUST EQUAL
      IF (INITM.EQ.1) GO TO 6
      LNCAT = 1
      GO TO 8
      CONTINUE
      LNCAT = NOSUB2
      CALL GETST (INUNIT, INFILE, ARRAY (MEANS1), DUM, NOSUB2, SURVEC, NOCHAN
      * , CHNVC, ARRAY (TTOP), COVAR, 0)
      LNCAT = NOSUB2
      GO TO 8
      CONTINUE
      IF (ISOKEY.EQ.1) CALL RDFILE (ARRAY (MEANS1), ARRAY (TTOP))
      CONTINUE
      IF (NOPTS.GT.0) GO TO 10
      WRITE (6,100) MAXDIM
100   FORMAT (' DIMENSION LIMITS EXCEEDED IN ISOCLS BY', 16,
      * ' REDUCE CHANNELS OR MAX. CLUSTERS')
      CALL CMERR
      CONTINUE
      IRD = TOTPTS/NOPTS
      IF (MOD (TOTPTS, NOPTS).EQ.0) GO TO 20
      KPTS = MOD (TOTPTS, NOPTS)
      IRD = IRD + 1
      IF (IRD.EQ.1) NOPTS = KPTS
      GO TO 25
20   KPTS = NOPTS
25   CONTINUE
      CALL ISODAT TO PERFORM CLUSTERING
      A1 = 1
      A2 = A1 + MAXCLS*NOFEAT
      CLD1 = A2 + MAXCLS*NOFEAT
      KPLCE = NOPTS*NOFEAT + IDAT1
      CALL ISODAT (ARRAY (IDAT1), ARRAY (KPLCE), ARRAY (MEANS1), ARRAY (N1),
      * ARRAY (STDEV1), KVAR (CLD1), ARRAY (FD1), KVAR (A1),
      * KVAR (A2))
      CHAIN CLUSTERS WHOSE DISTANCES ARE LESS THAN DLMIN
      LNCAT = LNCAT + DODU
      IF (ICHN.GT.0) CALL CHAIN (KVAR (CLD1))
      PRINT FINAL RESULTS
      CALL PRINT (-1, ARRAY (KPLCE), ARRAY (MEANS1), ARRAY (STDEV1),
      * KVAR (CLD1), ARRAY (FD1), ARRAY (N1))
      CREATE MAP OUTPUT TAPE FOR PMIS DAS IF DESIRED
      IF (MAPFMT.GT.0) CALL DSTAPE (ARRAY (KPLCE), KVAR (1), ARRAY (MEANS1),
      * ARRAY (FD1))
      LNCAT = LNCAT - DODU

```

15001790
 15001800
 15001810
 15001820
 15001830
 15001840
 15001850
 15001860
 15001870
 15001880
 15001890
 15001900
 15001910
 15001920
 15001930
 15001940
 15001950
 15001960
 15001970
 15001980
 15001990
 15002000
 15002010
 15002020
 15002030
 15002040
 15002050
 15002060
 15002070
 15002080
 15002090
 15002100
 15002110
 15002120
 15002130
 15002140
 15002150
 15002160
 15002170
 15002180
 15002190
 15002200
 15002210
 15002220
 15002230
 15002240
 15002250
 15002260
 15002270
 15002280
 15002290
 15002300
 15002310
 15002320
 15002330
 15002340
 15002350
 15002360
 15002370
 15002380
 15002390
 15002400
 15002410
 15002420
 15002430
 15002440
 15002450
 15002460
 15002470
 15002480
 15002490
 15002500
 15002510
 15002520
 15002530
 15002540

FILE IS0CLS

```

C*   CALCULATE COVARIANCE MATRIX FOR EACH CLUSTER
C*   IF (VARSIZ*LNCAT.GT.KVARDM)GO TO 30
    CALL COVAR1(KVAR,ARRAY(IDAT1),ARRAY(KPLCE),ARRAY(MEANS1),
    *   ARRAY(N1),IBAD)
C   CHECK FOR AT LEAST ONE SUBCLASS DELETED FOR SINGULAR MATRIX
C   IF (IBAD.NE.0)STOP=0
    IF (IBAD.NE.0)GO TO 25
C*
DO 26 II=1,LNCAT
26  NN(TOTSUB+II) = ARRAY(N1+II-1)
    TOTSUB = TOTSUB + LNCA
    NOCLS = NOCLS + 1
    TOTFLD = TOTFLD + NOFLD
    TOTVRT = TOTVRT + NVRT
    ARRAY(FD1+1)=IPT + FD1
    ARRAY(FD1+2)=LNCAT
    ARRAY(FD1+3)=NOFLD
C*
C*   WRITE STATS FOR THESE CLUSTERS ON SCRATCH FILE 18
C*
    IF (NOCLS.EQ.1) ADRES=BEGIN3
    IN=NOFEAT*LNCAT
    CALL RWRITE(ADRES,ARRAY(MEANS1),IN,JSTAT)
    ADRES=ADRES+IN
    IN=VARSIZ*LNCAT
    CALL RWRITE(ADRES,KVAR,IN,LSTAT)
    ADRES=ADRES+IN
C*   WAIT FOR I/O COMPLETION
60  IF (LSTAT.EQ.1) GO TO 60
C*
C*   GO READ IN ANOTHER CLASS
C*
    CORBAS=CORBAS+IPT
    IF (LAST.NE.1)GO TO 5
    IF (NOCLS.LT.NCLASS)GO TO 1
C*
C*   NOW READ SCRATCH FILE AND STORE ON SAVTAP FILE AND PUNCH ON
C*   CARDS IF REQUESTED.
C*
    FLD1 = 1
    VERTX1 = FLD1 + TOTFLD*4
    CLSNM1 = VERTX1 + TOTVRT*2
    NOSUB1 = CLSNM1 + NOCLS
    SUBNM1 = NOSUB1 + NOCLS
C   RETRIEVE INFORMATION FROM 'ARRAY'
C
    CALL GETINF(ARRAY(1),KVAR(FLD1),KVAR(VERTX1),KVAR(CLSNM1),
    *   KVAR(NOSUB1),KVAR(SUBNM1),NOCLS,TOTSUB)
C
    SWTCH = 1
C   OUTPUT STATS
C
    CALL LARMAN(SAVTAP,STAFIL,NOCLS,TOTSUB,NOFEAT,TOTFLD,TOTVRT,
    *   FETVEC,KVAR(FLD1),KVAR(VERTX1),KVAR(CLSNM1),KVAR(NOSUB1),
    *   KVAR(SUBNM1),NN,REGIN3,VARSIZ,PUNCH,DUMMY,STATKY,SWTCH)
    RETURN
30  KV=KVARDM
    WRITE(6,200)KV
    CALL CMERR
200  FORMAT(' DIMENSION LIMIT OF ',16,' FOR COVARIANCES EXCEEDED')
    RETURN
    END

```

IS002550
 IS002560
 IS002570
 IS002580
 IS002590
 IS002600
 IS002610
 IS002620
 IS002630
 IS002640
 IS002650
 IS002660
 IS002670
 IS002680
 IS002690
 IS002700
 IS002710
 IS002720
 IS002730
 IS002740
 IS002750
 IS002760
 IS002770
 IS002780
 IS002790
 IS002800
 IS002810
 IS002820
 IS002830
 IS002840
 IS002850
 IS002860
 IS002870
 IS002880
 IS002890
 IS002900
 IS002910
 IS002920
 IS002930
 IS002940
 IS002950
 IS002960
 IS002970
 IS002980
 IS002990
 IS003000
 IS003010
 IS003020
 IS003030
 IS003040
 IS003050
 IS003060
 IS003070
 IS003080
 IS003090
 IS003100
 IS003110
 IS003120
 IS003130
 IS003140
 IS003150
 IS003160
 IS003170
 IS003180
 IS003190
 IS003200
 IS003210

●●●●●

CSEND

C.

● ● ● ●

45

1000

1

COV000010
COV000020
COV000030
COV000040
COV000050
COV000060
COV000070
COV000080
COV000090
COV000100
COV000110
COV000120
COV000130
COV000140
COV000150
COV000160
COV000170
COV000180
COV000190
COV000200
COV000210
COV000220
COV000230
COV000240
COV000250
COV000260
COV000270
COV000280
COV000290
COV000300
COV000310
COV000320
COV000330
COV000340
COV000350
COV000360
COV000370
COV000380
COV000390
COV000400
COV000410
COV000420
COV000430
COV000440
COV000450
COV000460
COV000470
COV000480
COV000490
COV000500
COV000510
COV000520
COV000530
COV000540
COV000550
COV000560
COV000570
COV000580
COV000590
COV000600
COV000610
COV000620
COV000630
COV000640
COV000650
COV000660
COV000670
COV000680
COV000690
COV000700
COV000710
COV000720
COV000730
COV000740
COV000750
COV000760

FILE: COVAR1

50	CONTINUE	COV00770
	IACEPT=PMIN+NOFEAT	COV00780
	IF(IACEPT.LT.NOFEAT)GO TO 58	COV00790
C	CHECK FOR SINGULAR COVARIANCE MATRIX	COV00800
C	DO 51 I=1,LNCAT	COV00810
	CALL CHLOET(COVAR(1,I),NOFEAT,DUMM,DET)	COV00820
	IF(DET.LT.TOL)GO TO 52	COV00830
51	CONTINUE	COV00840
	GO TO 58	COV00850
C	DELETE SINGULAR COVARIANCE MATRIX CLUSTER	COV00860
C	WRITE(6,160)I	COV00870
52	IF(LNCAT.EQ.1)CALL CMERR	COV00880
	IRAD=1	COV00890
	LNCAT=LNCAT-1	COV00900
	LLNCAT=LLNCAT-1	COV00910
	DO 53 II=1,LNCAT	COV00920
	DO 53 III=1,NOFEAT	COV00930
	MEANS(III,II)=MEANS(III,II+1)	COV00940
53	CONTINUE	COV00950
58	RETURN	COV00960
160	FORMAT(2X,'CLUSTER',IS,' DELETED FOR SINGULARITY')	COV00970
C	IF(STATKY.NE.1)RETURN	COV00980
55	WRITE(6,HEAD)	COV00990
	WRITE(6,150)CLSNAM	COV01000
	DO 80 I=1,LNCAT	COV01010
	WRITE(6,90)I	COV01020
	DO 70 LOC=1,NOFEAT,12	COV01030
	ISTOP=LOC+11	COV01040
	IF(ISTOP.GT.NOFEAT)ISTOP=NOFEAT	COV01050
	WRITE(6,140)(CH,FETVEC(J),J=LOC,ISTOP)	COV01060
	II=1	COV01070
	KINC=1	COV01080
	DO 60 J=LOC,NOFEAT	COV01090
	K=J*(J+1)/2-II+1	COV01100
	JK=K+KINC-1	COV01110
	WRITE(6,100)(COVAR(M,I),M=K,JK)	COV01120
	II=II+1	COV01130
60	IF(KINC.LT.ISTOP.AND.KINC.LT.12)KINC=KINC+1	COV01140
	WRITE(6,110)	COV01150
70	CONTINUE	COV01160
80	CONTINUE	COV01170
	RETURN	COV01180
90	FORMAT(///' COVARIANCE MATRIX FOR CLUSTER',I4//)	COV01190
100	FORMAT(/6X,12F9.2)	COV01200
110	FORMAT(///)	COV01210
120	FORMAT(1H1)	COV01220
140	FORMAT(9X,12(A3,I2,'')*.3X))	COV01230
150	FORMAT(/' COVARIANCES FOR CLASS',2X,A4//)	COV01240
	END	COV01250
		COV01260
		COV01270
		COV01280
		COV01290

FILE: ISODAT

```

SUBROUTINE ISODAT(C,IPLACE,MEANS,N,STDEV,CLD,FLDINF,AVP,AMN)
  IMPLICIT INTEGER (A-Z)
  IMPLICIT INTEGER (A-Z)
  INCLUDE COMBKS,LIST
  INCLUDE COMBK6,LIST
C
CMS360
C
  INCLUDE CMRK16,LIST
  COMMON/PASS/STOP,LNCAT,NMIN,KRN,STDMAX,DLMIN,SEP,
  * MAP,SPTRIG,IRD,KPTS,NOPTS,PUNCH,
  * ICHN,CHNTHS,ICHAIN(62),NWDS,IREGIN,REGIN1,
  * REGIN2,BEGIN3,CLSNAM,NOFLD,IPT,TOTWRD,TOTPTS,
  * NCLASS,NOCLS,TOTSUR,TOTFLD,TOTVRT,NOCL,NVRT
  * ,NXTCLS,NOFEAT,MAXCLS,FETVEC(30),SYMMTX(62)
  * ,VAPSI7,STATKY,ISOKEY,MAPFMT,MAPKEY,SEQUEN(20),PERCEN,SIMERP
  * ,IORDER,INUNIT,INFILE,INITM,PMIN,SUBVEC(62),NOSUB2,CHNVC(30)
  * ,NOCHAN,FRCOMP,NOSEQ,MEANDU,MEANDU,
  * SYMDO,SYMDO,ITRIG0,ITRIGU,DOFLAG,
  * DUFLAG,DODU,STDOTS(60),NSDOTS,SUNCOR(30),LLNCAT,
  * DVERT(250,2),DURECT(60,2),DVPNT(11,2),IDCNT(2),NDOU(2)
  * ,MXFFT1,MXPOP
  REAL SUNCOR
  COMMON/GLOBAL/HEAD(63),MAPTAP,DATEP,SAVTAP,BMFILE,BMKEY,
  * HISFIL,HISKEY,TRFORM,ERIPTP,ERPKEY,MAPUNT,NOFILE,
  * DRUMAD,DRMWDS,PAGSI7,DATFIL,STAFIL,ASAV,ASAVFL
  * ,NHSTUN,NHSTFI,SCTRUN,MAPFIL
  * ,DOTUNT,DOTFIL,NCHPAS,TRNSFL,BMTRFL,HISTFL,PCHUNT,
  * CRDUNT,PRUNT,RANDIO
  COMMON/ISOLNK/SUNANG(8),ISUNT,ISUNC,SMSTR,SMSTP,SMINC,LINSKP
CSEND
  EQUIVALENCE (SGMIN,STDMAX)
  REAL MEANS,STDEV,STDMAX,SEP,AVP,C,AMN,SGMA,RND,
  * TEST,DMIN,DLMIN,CLD,TIME,PERCEN,DIJ
  REAL ESUM,ESOT,MEAN(30,62),SDIJ
  LOGICAL DEL
  DIMENSION AVP(NOFEAT,MAXCLS),ISGMA(62)
  DIMENSION C(NOFEAT,NOPTS),IPLACE(NOPTS)
  DIMENSION AMN(NOFEAT,MAXCLS),SGMA(62)
  DIMENSION MEANS(NOFEAT,MAXCLS),N(MAXCLS)
  DIMENSION STDEV(NOFEAT,MAXCLS),CLD(MAXCLS,MAXCLS)
  DIMENSION FLDINF(1)
  REAL SQUM
  DIMENSION PTR(62)
  DATA SS/'S'/'CC'/'C'/'
  EQUIVALENCE (KDIM,NOFEAT),(LNCAT,INCAT)
  DEL=.FALSE.
  ISFQ=0
  MAXCL=MAXCLS-DODU
  IDUM=LNCAT-DODU-MAXCLS
  IF (IDUM.GT.0) LNCAT=LNCAT-IDUM
  ISTOP=STOP
  SPLFIN=0
  KKT=1
  DO 5 I=1,30
    SUNCOR(I)=1.
  5 IF (ISUNC.NE.0.OR.ISUNT.NE.0) CALL SUNFAC(SUNCOR,SUNANG,
  * FETVEC,NOFEAT,ISUNC,ISUNT)
C
  LK=K
C*
  ASSIGN DATA TO CLUSTERS
C*
  10 CONTINUE
  LNCAT=LNCAT+DODU
  IF (DOFLAG.EQ.0) GO TO 12
  DO 11 J=1,NOFEAT
    MEANS(J,LNCAT+1)=MEANDU
  11 IF (DUFLAG.EQ.0) GO TO 14
  DO 13 J=1,NOFEAT
    MEANS(J,LNCAT)=MEANDU
  13 CONTINUE
  DO 15 K=1,LLNCAT
    DO 15 J=1,NOFEAT
      MEAN(J,K)=MEANS(J,K)
  15 IF (LNCAT.LE.1.AND.KKT.GT.1) GO TO 530
  CALL PSPLIT(MEANS,STDEV,N,CLD,C,IPLACE,AVP,AMN,MEANS)
  CALL CLOCK(TIME)
  IF (MOD(KKT,KRN).EQ.0) WRITE(6,120)KKT,TIME
C 120 FORMAT(' CUMULATIVE TIME AFTER ASSIGNING DATA TO CLUSTERS FOR ITER
C *ATION',I4,' IS',F10.6)

```

FILE: ISODAT

```

      IF (EPCOMP.NE.1) GO TO 135
      ESUM=0.0
      DO 132 J=1,NOFEAT
      DO 132 K=1,LNCAT
      FSUM=FSUM+N(K)*(STDEV(J,K))**2/TOTPTS
132  CONTINUE
      ESQT=SQRT(ESUM/NOFEAT)
      WRITE(6,133) ESQT,PERCEN,STDMAX
133  FORMAT(1X,///' ERCOMP= ',F7.3,' PERCEN = ',F5.3,' STDMAX = ',
      *F7.3/)
C*
C*   CALCULATE DISTANCES BETWEEN CLUSTER CENTERS
C*
135  CALL CLDIST(CLD,STDEV,MEANS)
C*
C*   IF STOP EQUALS ZERO DELETE SMALL CLUSTERS
C*
      LNCAT=LLNCAT
      IF (MOD(KKT,MAP)) 150,140,150
140  CALL PRINT(KKT,IPLACE,MEANS,STDEV,CLD,FLDINF,N)
      GO TO 161
150  IF (MOD(KKT,KRN)) 161,160,161
160  CONTINUE
      CALL PRINT(KKT,IPLACE,MEANS,STDEV,CLD,FLDINF,N)
161  CONTINUE
      LNCAT=LLNCAT-DODU
      IF (ISTOP.EQ.0) GO TO 162
C
C   FOR ITERATION N CHECK N(K) AGAINST PMIN + NOFEAT
C
      IF (ISEQ.NE.NOSEQ) GO TO 169
      ISTOP = 0
162  DO 163 K = 1,LNCAT
      IF (N(K) - (PMIN + NOFEAT)) 167,163,163
163  CONTINUE
      IF (.NOT.DEL) RETURN
      DO 164 KK=1,LLNCAT
      DO 164 KKK=1,NOFEAT
164  MEANS(KKK,KK) = MEAN(KKK,KK)
      CALL PSPLIT(MEANS,STDEV,N,CLD,C,IPLACE,AVP,AMN,MEANS)
      DO 165 KK=1,LLNCAT
      DO 165 KKK=1,NOFEAT
165  MEAN(KKK,KK) = MEANS(KKK,KK)
      CALL CLDIST(CLD,STDEV,MEANS)
      RETURN
167  WRITE(6,168) K,N(K),PMIN,NOFEAT
168  FORMAT(/' CLUSTER ',13,' REMOVED FOR HAVING ONLY ',16,' POINTS. '/
      *' MIN. POINTS IS ',14,' * ',12,'))
      RETF=1
      LK=K
      GO TO 570
171  K=LK
      DEL = .TRUE.
      GO TO 162
169  CONTINUE
170  CONTINUE
C
C   ON ITERATIONS 1 THRU N-1 CHECK N(K) AGAINST NMIN
C
      DO 180 K=1,INCAT
      IF (N(K)-NMIN) 190,180,180
180  CONTINUE
      IF (DEL) CALL CLDIST(CLD,STDEV,MEANS)
      GO TO 220
190  IF (MOD(KKT,KRN)) 200,195,200
195  WRITE(6,210) K,N(K),NMIN
200  RETF=2
      LK=K
      GO TO 570
201  K=LK
      DEL = .TRUE.
      GO TO 170
210  FORMAT(/' CLUSTER ',12,' REMOVED FOR HAVING ONLY ',16,'
      *' ELEMENTS. MIN. NO. ELEMENTS IS ',16)
220  CONTINUE
C
C*   SPLIT ITERATION

```

5000800
 5000810
 5000820
 5000830
 5000840
 5000850
 5000860
 5000870
 5000880
 5000890
 5000900
 5000910
 5000920
 5000930
 5000940
 5000950
 5000960
 5000970
 5000980
 5000990
 5001000
 5001010
 5001020
 5001030
 5001040
 5001050
 5001060
 5001070
 5001080
 5001090
 5001100
 5001110
 5001120
 5001130
 5001140
 5001150
 5001160
 5001170
 5001180
 5001190
 5001200
 5001210
 5001220
 5001230
 5001240
 5001250
 5001260
 5001270
 5001280
 5001290
 5001300
 5001310
 5001320
 5001330
 5001340
 5001350
 5001360
 5001370
 5001380
 5001390
 5001400
 5001410
 5001420
 5001430
 5001440
 5001450
 5001460
 5001470
 5001480
 5001490
 5001500
 5001510
 5001520
 5001530
 5001540
 5001550
 5001560
 5001570
 5001580

FILE: ISODAT

```

C
C
225 DO 225 I=1,INCAT
    PTR(I)=1
    ISPLT=0
    DO 260 K=1,INCAT
C*
C* FIND MAXIMUM STANDARD DEVIATION PER CLUSTER
C*
        SGMA(K) = 0.
        DO 250 J = 1,NOFEAT
            SDUM = STDEV(J,K)*SUNCOR(J)
            IF (SDUM - SGMA(K)) 250,240,240
240         ISGMA(K) = J
            SGMA(K) = SDUM
250        CONTINUE
        IF (SGMA(K).GE.STDMAX) ISPLT=ISPLT+1
260        CONTINUE
        IF (2*LNLCAT.GT.MAXCL) CALDESCEN(SGMA,LNLCAT,ISGMA,PTR)
        TEST=FLOAT(ISPLT)/FLOAT(LNLCAT)
        IF (TEST.LE.PERCEN) SPLFIN=1
        IF (KKT.GT.ISTOP) SPLFIN = 1
        IF (SPLFIN.EQ.0) GO TO 270
        IF (MOD(KKT,KRN).EQ.0) WRITE(6,503)
503        FORMAT( /)
        IF (MOD(KKT,KRN).EQ.0) WRITE(6,502)
502        FORMAT(1X,'USER INPUT*SPLIT-COMBINE SEQUENCE OF ITERATIONS')
        ISFO=ISEQ+1
        IF (SEQUEN(ISEQ).EQ.SS) GO TO 270
        IF (SEQUFN(ISEQ).EQ.CC) GO TO 410
C
C IS SPLITTING REQUIRED
C
270 K=1
    NCAT=INCAT
280 IF (K-NCAT) 290,290,500
290 IF (STDMAX-SGMA(K)) 300,300,310
300 IF (N(K)-(NMIN+NMIN+2)) 310,310,320
310 K=K+1
    GO TO 280
C
C SPLIT CLUSTER K
C
320 TRIG1=1
    DEL=.TRUE.
    KX=ISGMA(K)
330 INCAT=INCAT+1
    LLNCAT=LLNCAT+1
    IF (LLNCAT.LE.MAXCLS) GO TO 350
    IF (MOD(KKT,KRN).EQ.0) WRITE(6,340) KKT
340 FORMAT( /' MAXIMUM CLUSTERS ON ITERATION',14/ ' SPLITTING REQUIRED B
    *UT NOT PERFORMED' /)
    LNCAT = MAXCL
    LLNCAT=MAXCLS
    GO TO 500
350 INC=INCAT
    LL=PTR(K)
360 DO 370 I=1,KDIM
370     AMN(I,INC)=AMN(I,LL)
380     AMN(KX,LL)=AMN(KX,LL)+SEP*SGMA(K)
        AMN(KX,INC)=AMN(KX,INC)-SEP*SGMA(K)
        SGMA(K)=0.0
        IF (MOD(KKT,KRN)) 400,401,400
401     WRITE(6,390) LL,KX,INC
390     FORMAT('0 CLUSTER ',12,' IS SPLIT IN THE ',12,'TH PARAMETER INTO C
        CLUSTER ',12)
400     CONTINUE
        K=K+1
        GO TO 280
C
C EVEN ITERATION
C
C ARE CLUSTERS TO BE COMBINED
C
410 CONTINUE
    DO 405 L=1,LNCAT
405     PTR(L)=1
C
    NOCOMB=0

```

FILE: ISODAT

```

      NOCLST=LNCAT-1
      L=-1
406  L=L+2
      IF (L.GT.NOCLST) GO TO 480
      NOCLTR = LNCAT - 1
      KK=0
      DMIN=DLMIN
      DO 430 I=1,NOCLTR
C      IF (PTR(I).EQ.0) GO TO 430
      II=I+1
      DO 425 J=II,LNCAT
      IF (PTR(J).EQ.0) GO TO 425
      SDIJ = 0.0
      DO 420 JJ=1,KDIM
      SDIJ=SDIJ+((AMN(JJ,I)-AMN(JJ,J))*2/(STDEV(JJ,I)*STDEV(JJ,J)))
420  CONTINUE
      DIJ=SQRT(SDIJ)
C      IF (DIJ.GT.DMIN) GO TO 425
      DMIN=DIJ
      KK=I
      KKK=J
425  CONTINUE
430  CONTINUE
C      IF (KK.EQ.0) GO TO 480
      PTR(KK)=0
C      COMBINE CLUSTERS KK AND KKK
C      DEL=.TRUE.
      RND=1.0 /FLOAT(N(KK)+N(KKK))
C      DO 460 K=1,KDIM
      AMN(K,KK)=(N(KK)*AMN(K,KK)+N(KKK)*AMN(K,KKK))*RND
460  CONTINUE
      RETF=3
      LK=KKK
      GO TO 570
461  KKK=LK
      IF (KKK.EQ.(LNCAT+1)) GO TO 435
C      MOVE POINTERS UP
C      DO 175 K=KKK,LNCAT
      PTR(K) = PTR(K+1)
C      175  PTR(K) = PTR(K+1)
435  IF (MOD(KKT,KRN))440,441,440
441  WRITE (6,440)KK,KKK,KK
440  IF (L.LT.NOCLST) GO TO 406
C      CONTINUE
490  FORMAT(' CLUSTERS ',I2,' AND ',I2,' HAVE BEEN COMBINED INTO CLUST
      2ER ',I2)
C* REINITIALIZE
C*
500  CONTINUE
      DO 510 J=1,MAXCLS
      SGMA(J)=0.0
      ISGMA(J)=0
      DO 510 K=1,KDIM
      AVP(K,J)=0.0
      STDEV(K,J)=0.0
      MEANS(K,J)=AMN(K,J)
      AMN(K,J)=0.0
510  CONTINUE
      KKT=KKT+1
      DEL=.FALSE.
      GO TO 10
C      530 IF (KKT.NE.2) GO TO 550
      WRITE (6,540)
540  FORMAT(' THE ORIGINAL CLUSTER WAS NOT SPLIT - EXAMINE THE INPUT VA
      *LINE FOR STDMAX')
      KKT=1
      ISTOP=0
      GO TO 10

```

IS002380
 IS002390
 IS002400
 IS002410
 IS002420
 IS002430
 IS002440
 IS002450
 IS002460
 IS002470
 IS002480
 IS002490
 IS002500
 IS002510
 IS002520
 IS002530
 IS002540
 IS002550
 IS002560
 IS002570
 IS002580
 IS002590
 IS002600
 IS002610
 IS002620
 IS002630
 IS002640
 IS002650
 IS002660
 IS002670
 IS002680
 IS002690
 IS002700
 IS002710
 IS002720
 IS002730
 IS002740
 IS002750
 IS002760
 IS002770
 IS002780
 IS002790
 IS002800
 IS002810
 IS002820
 IS002830
 IS002840
 IS002850
 IS002860
 IS002870
 IS002880
 IS002890
 IS002900
 IS002910
 IS002920
 IS002930
 IS002940
 IS002950
 IS002960
 IS002970
 IS002980
 IS002990
 IS003000
 IS003010
 IS003020
 IS003030
 IS003040
 IS003050
 IS003060
 IS003070
 IS003080
 IS003090
 IS003100
 IS003110
 IS003120
 IS003130
 IS003140
 IS003150
 IS003160

FILE: ISODAT

```
550 WRITE (6,560)KKT
560 FORMAT(///, AFTER '.14.' ITERATIONS ALL DATA HAS BEEN ASSIGNED TO 0
      *NE CLUSTER./)
      KKT=1
      ISTOP=0
      GO TO 10
570 CONTINUE
C*
C* ROUTINE TO DELETE A CLUSTER
C*
      INCAT=INCAT-1
      LLNCAT=LLNCAT-1
      IF (LK.EQ.(INCAT+1).AND.DODU.EQ.0) GO TO (171,201,461),RETF
      DO 561 J=LK,LLNCAT
      DO 552 L=1,KDIM
      AMN(L,J)=AMN(L,J+1)
      MEANS(L,J)=MEANS(L,J+1)
      MEAN(L,J)=MEAN(L,J+1)
552 STDEV(L,J)=STDEV(L,J+1)
      N(J)=N(J+1)
561 CONTINUE
      GO TO (171,201,461),RETF
      END
```

5003170
5003180
5003190
5003200
5003210
5003220
5003230
5003240
5003250
5003260
5003270
5003280
5003290
5003300
5003310
5003320
5003330
5003340
5003350
5003360
5003370
5003380
5003390

FILE: PSPLIT

```

SUBROUTINE PSPLIT(MEANS,STDEV,N,CLD,C,IPLACE,AVP,AMN,MEN)
IMPLICIT INTEGER (A-Z)

C
C
C   INCLUDE COMRK5.LIST
C   INCLUDE CMBK16.LIST
COMMON/PASS/STOP,LNCAT,NMIN,KRN,STOMAX,DLMIN,SEP,
MAP,SPTRIG,IRD,KPTS,NOPTS,PUNCH,
• ICHN,CHNTHS,ICHAIN(62),NWDS,IHGIN,REGIN1,
• REGIN2,REGIN3,CLSNAM,NOFLD,IP1,TOTWRD,TOTPTS,
• NCLASS,NOCLS,TOTSUN,TOTFLD,TOTVRT,NOCL,NVRT
• ,NXTCLS,NOFEAT,MAXCLS,FETVEC(30),SYMMTX(62)
• ,VARS17,STATKY,ISOKEY,MAPFMT,MAPKEY,SEQUEN(20),PERCEN,SIMERP
• ,IORDER,INUNIT,INFILE,INITM,PMIN,SUBVEC(62),NOSUB2,CHNVC(30)
• ,NOCHAN,FRCOMP,NOSEQ,MEANDU,MEANDU,
• SYMDO,SYMDOU,ITRIG,ITRIGU,DOFLAG,
• DUFLAG,DODU,STDOTS(60),NSDOTS,SUNCOR(30),LLNCAT,
• DVERT(250,2),DRECT(60,2),DVPNT(11,2),IDCNT(2),NDOU(2)
• ,MAXFT1,MAXPOP
REAL SUNCOR
COMMON/ISOLNK/SUNANG(8),ISUNT,ISUNC,SMSTR,SMSTP,SMINC,LINSKP
CSEND
DIMENSION C(NOFEAT,NOPTS),IPLACE(NOPTS),AMN(NOFEAT,MAXCLS)
DIMENSION STDEV(NOFEAT,MAXCLS),CLU(MAXCLS,MAXCLS),N(MAXCLS)
DIMENSION AVP(NOFEAT,MAXCLS),MEANS(NOFEAT,MAXCLS)
REAL MEN(NOFEAT,MAXCLS)
REAL AMN,STDEV,AVP,SU1ST,DIST,C,RND,MEANS
DIMENSION CSUN(30)
REAL CSUN
REAL DUM,DUMA
DUM = .00001

C
IF(DOFLAG.NE.0) N(LNCAT+1)=NDOU(DOFLAG)
IF(DUFLAG.NE.0) N(LNCAT+DODU)=NDOU(DODU)
DO 5 I=1,LNCAT
N(I)=0
DO 5 J=1,NOFEAT
AMN(J,I)=0.0
STDEV(J,I)=0.0
AVP(J,I)=0.0

S
C
C
C   ASSIGN DATA TO CLUSTERS
ADRES1=REGIN1
ADRES2=REGIN2
ICCT=NOPTS
IRC=IRD
20 IF(IRC.EQ.1) ICCT=KPTS
IF(IRC.EQ.0) GO TO 40
IWRDS=NOFEAT*ICCT
CALL READ(ADRES1,C,IWRDS,ISTAT)
ADRES1=ADRES1+IWRDS
25 IF(ISTAT.EQ.1) GO TO 25
IF(ISTAT.EQ.0) GO TO 40
IF(ISTAT.EQ.2) GO TO 40
WRITE(6,30) ISTAT
30 FORMAT(' ERROR READING DRUM---ISTAT=',I4)
40 CONTINUE
IF (ISUNT.EQ.0.AND.ISUNC.EQ.0) GO TO 50
DO 49 I=1,ICCT
IF(DODU.EQ.0) GO TO 42
DO 41 K=1,NOFEAT
CDUM = C(K,I)
IF (CDUM.NE.MEANDU.AND.CDUM.NE.MEANDU) GO TO 42
41 CONTINUE
IF (CDUM.EQ.MEANDU) IPLACE(I) = LNCAT + 1
IF (CDUM.EQ.MEANDU) IPLACE(I) = LNCAT + DODU
GO TO 49
42 KK=1
SDIST=10.0E+20
DO 46 J=1,LNCAT
DIST=0.
DO 44 K=1,NOFEAT
CSUN(K)=C(K,I)
44 DIST=DIST+ABS(MEANS(K,J)-CSUN(K))*SUNCOR(K)
IF (DIST - SDIST) 45,46,46
45 KK=J
SDIST=DIST
46 CONTINUE
47 CONTINUE

```

PSP00010
PSP00020
PSP00030
PSP00040
PSP00050
PSP00060
PSP00070
PSP00080
PSP00090
PSP00100
PSP00110
PSP00120
PSP00130
PSP00140
PSP00150
PSP00160
PSP00170
PSP00180
PSP00190
PSP00200
PSP00210
PSP00220
PSP00230
PSP00240
PSP00250
PSP00260
PSP00270
PSP00280
PSP00290
PSP00300
PSP00310
PSP00320
PSP00330
PSP00340
PSP00350
PSP00360
PSP00370
PSP00380
PSP00390
PSP00400
PSP00410
PSP00420
PSP00430
PSP00440
PSP00450
PSP00460
PSP00470
PSP00480
PSP00490
PSP00500
PSP00510
PSP00520
PSP00530
PSP00540
PSP00550
PSP00560
PSP00570
PSP00580
PSP00590
PSP00600
PSP00610
PSP00620
PSP00630
PSP00640
PSP00650
PSP00660
PSP00670
PSP00680
PSP00690
PSP00700
PSP00710
PSP00720
PSP00730
PSP00740
PSP00750
PSP00760
PSP00770
PSP00780
PSP00790


```

N(KK)=N(KK)+1
IPLACE(I)=KK
DO 48 K=1,NOFEAT
AMN(K,KK)=AMN(K,KK)+CSUN(K)
AVP(K,KK)=AVP(K,KK)+CSUN(K)**2
48 CONTINUE
49 CONTINUE
GO TO 101
50 DO 100 I = 1,ICCT
KK=1
IF(DODU.EQ.0) GO TO 52
DO 51 K=1,NOFEAT
CDUM=C(K,I)
IF (CDUM.NE.MEAND0.AND.CDUM.NE.MEANDU) GO TO 52
51 CONTINUE
IF (CDUM.EQ.MEAND0) IPLACE(I) = LNCAT + 1
IF (CDUM.EQ.MFANDU) IPLACE(I) = LNCAT + DODU
GO TO 100
52 CONTINUE
KK = 1
SDIST=10.0E+20
DO 70 J=1,LNCAT
DIST=0.0
DO 55 K = 1,NOFEAT
CSUN(K) = C(K,I)
55 DIST = DIST + ARS(MEANS(K,J) - CSUN(K))
60 IF (DIST-SDIST)60,70,70
KK=J
70 SDIST=DIST
70 CONTINUE
80 CONTINUE
N(KK)=N(KK)+1
IPLACE(I)=KK
DO 90 K=1,NOFEAT
AMN(K,KK) = AMN(K,KK) + CSUN(K)
90 AVP(K,KK) = AVP(K,KK) + CSUN(K)**2
100 CONTINUE
101 CONTINUE
IF(IRC.EQ.0)GO TO 110
CALL RWRITE(ADRES2,IPLACE,ICCT,ISTAT)
ADRES2=ADRES2+ICCT
105 IF(ISTAT.EQ.1)GO TO 105
110 IRC=IRC+1
IF(IRC.GT.0)GO TO 20
KA = 1
115 CONTINUE
DO 130 K=KA,LNCAT
IF(N(K).EQ.0)GO TO 130
RND=FLOAT(N(K))
DO 130 J=1,NOFEAT
AMN(J,K)=AMN(J,K)/RND
MEANS(J,K)=AMN(J,K)
STDEV(J,K)=SORT(AVP(J,K)/RND-AMN(J,K)*AMN(J,K))
DUMA = STDEV(J,K)
130 IF (DUMA.LT.DUM) STDEV(J,K) = DUM
CONTINUE
RETURN
END

```

P\$P000800
 P\$P000810
 P\$P000820
 P\$P000830
 P\$P000840
 P\$P000850
 P\$P000860
 P\$P000870
 P\$P000880
 P\$P000890
 P\$P000900
 P\$P000910
 P\$P000920
 P\$P000930
 P\$P000940
 P\$P000950
 P\$P000960
 P\$P000970
 P\$P000980
 P\$P000990
 P\$P001000
 P\$P001010
 P\$P001020
 P\$P001030
 P\$P001040
 P\$P001050
 P\$P001060
 P\$P001070
 P\$P001080
 P\$P001090
 P\$P001100
 P\$P001110
 P\$P001120
 P\$P001130
 P\$P001140
 P\$P001150
 P\$P001160
 P\$P001170
 P\$P001180
 P\$P001190
 P\$P001200
 P\$P001210
 P\$P001220
 P\$P001230
 P\$P001240
 P\$P001250
 P\$P001260
 P\$P001270
 P\$P001280
 P\$P001290
 P\$P001300
 P\$P001310
 P\$P001320
 P\$P001330
 P\$P001340
 P\$P001350
 P\$P001360
 P\$P001370
 P\$P001380

FILE: R0DATA

```

C* THIS SUBROUTINE COORDINATES THE ROUTINES TO READ FIELDS OF DATA FROM THE IMAGE TAPE AND STORE IT ON A DRUM FILE FOR THE ISOCLS ROUTINES.
C*
C* SUBROUTINE R0DATA (ARRAY, TOP, IDATA, IDIM, LAST)
C* IMPLICIT INTEGER (A-Z)
C* DIMENSION ARRAY (TOP), FLDINF (6), IDATA (IDIM), FL (12), LSTAT (3)
C* INCLUDE CO4HKS.LIST
C* INCLUDE CO4HKS.LIST
C* COMMON /PASS/STOP, LNCAT, NMIN, KRN, STDMAX, DLMIN, SEP,
C* MAP, CPT, ITR, KPTS, NOPTS, PUNCH,
C* ICHN, CHNTHS, ICHAIN (62), NCHS, ICHN, ICHN1,
C* HEGIN2, HEGIN, CLSHAM, NOFLD, IPT, TOTWRD, TOTPTS,
C* NCLASS, NOCLS, TOTSUM, TOTFLD, TOTVRT, NOCL, NVRT,
C* NITCLS, NOFEAT, MARCLS, IFTV, C (30), SYMTR (62),
C* VANSIZ, STATKY, ISOKEY, MAPENT, MAPKEY, SEQUEN (20), PERCEN, SIMERP,
C* IOWFR, IOWITT, INFILE, INITM, PMIN, SUBVEC (62), NOSUB2, CHNVC (30),
C* NOCHAM, ERCONP, NOSEQ, MEAND, MEANDU,
C* SYMDO, SYMDO1, ITRIG, ITRIGU, NOFLAG,
C* DUEFLAG, DODU, STDOTS (60), NSDOTS, SINCON (30), LLNCAT,
C* DVNT (256, 2), DRECT (60, 2), DVNT (11, 2), IDCNT (2), NDOU (2),
C* NXFFT1, MAXPOP,
C* REAL SINCON
C* COMMON /GLOBAL/HEAD (63), MAPTAP, DATAP, SAVTAP, HMFILE, HMKEY,
C* HISFIL, HISKEY, TRFOM, EXPT, EMPKEY, MAPUNT, NOFILE,
C* PRIMAD, DRWDS, PAGESZ, DATFIL, STAFIL, ASAV, ASAVFL,
C* MCHSTN, MCHST1, SCTRUN, MAPFIL,
C* DOTINT, DOTFIL, NCHPAS, TRNSFL, HMTFL, MISTFL, PCHUNT,
C* CROUNT, PRUNT, RANDIO
C*
C* PRUNT = 30
C* DIMENSION CARD (20)
C* EQUIVALENCE (FLDINF (1), LINST), (FLDINF (4), SAMSTR),
C* (FLDINF (2), LINFND), (FLDINF (5), SAMEND),
C* (FLDINF (3), LININC), (FLDINF (6), SAMINC)
C* DATA LPWN / 1 /
C* DIMENSION L (2), LDOU (11, 2), ID (35), ID (35), MDINT (11, 2),
C* DINT (12, 2), DPINT (12, 2), DIN (70)
C* DATA DNAME / 10 THE /
C* DATA DNAME / UNID /
C* DIMENSION FLOSAV (4, 10), VF4TEX (220)
C*
C* RESERVE 2000 LOCATIONS OF 'ARRAY' FOR FIELD DEFINITION INFORMATION.
C* THE REMAINDER OF 'ARRAY' IS USED FOR I/O BUFFERS.
C*
C* CLASS AND FIELD INFORMATION STORED AS FOLLOWS
C*
C* ARRAY (1) = CLASS NAME
C* ARRAY (2) = RESERVED FOR INDEX POINTER TO NEXT CLASS NAME
C* ARRAY (3) = RESERVED FOR NO. OF CLUSTERS IN THIS CLASS
C* ARRAY (4) = NO. OF FIELDS FOR THIS CLASS
C* ARRAY (5) = FIRST FIELD NAME FOR THIS CLASS
C* (6) = NO. OF VERTICES FOR THIS FIELD (INV)
C* (7) - (7+NV*2) = ACTUAL VERTEX NUMBERS
C* (8+NV*2) = TOTAL PTFELS IN THIS FIELD
C* (9+NV*2) - (10+NV*2) = FLDINF BLOCK FOR THIS FIELD
C*
C* CALL TAPHDR (DATAP, DATFIL)
C*
C* 1 CONTINUE
C* RESERV=2000
C* ADDRFS=BEGIN1
C* IWRD=1
C* IWRD=0
C* NVRT=0
C* LAST=0
C* TOTWRD=0
C* IDP = 0
C* DOFLAG = 0
C* DUEFLAG = 0
C* DODU = 0
C* NDOU (1) = 0
C* NDOU (2) = 0
C*
C* 2 REINDEX=RESERV+1
C* NRUES=3
C* MAXDIN=TOP-RESERV
C* HUESIZ=MAXDIN/2 (NRUES+DOFEAT) * NOFEAT
C* IF (HUESIZ.GT. 100) GO TO 3
C* RESERV=RESERV+100
C* IF (RESERV.GT. 30) GO TO 2

```

FILE: RDDATA

```

      GO TO 70
3  CONTINUE
   NOFLD=0
   IPT=1
   TOTVT2=0
   IF (NOCL.EQ.0) GO TO 5
4  ARRAY(IPT)=NXTCLS
   IPT=IPT+4
   WRITE(6,HEAD)
   WRITE(6,500)NXTCLS
C*
C*  READ A FIELD DESCRIPTION FROM CARDS.
C*
5  ICK = LARFAD(ARRAY(IPT),ARRAY(IPT+2),FLD1,ARRAY(IPT+1))
   IF (ICK.NE.-3) GO TO 1000
   WRITE(6,140)
   READ (RRUNIT,150) (CARD(I), I=1,20)
   WRITE(6,160) (CARD(I), I=1,20)
150  FORMAT(20A4)
160  FORMAT(1X,20A4)
   REWIND RRUNIT
   IDP=IDP+1
   IDCNT(IDP)=0
   DVPNT(1,IDP)=1
   READ(30,100) DNAME
   REWIND 30
   IF (DNAME.EQ.DONAME) ITRIG0=1
   IF (DNAME.EQ.DUNAME) ITRIGU=1
   IF (DNAME.EQ.DONAME) IS=2
   IF (DNAME.EQ.DUNAME) IS=1
   INDV=1
   IDP=1
   GO TO 5
1000 IF (ICK.LE.0.OP.IDP.LE.0) GO TO 1030
   IF (IDCNT(IDP).LT.10) GO TO 1025
   WRITE(6,170)
170  FORMAT(// ' TOO MANY DO OR DU FIELDS THESE IGNORED*')
   GO TO 5
1025 CONTINUE
   READ (RRUNIT,150) (CARD(I), I=1,20)
   WRITE(6,160) (CARD(I), I=1,20)
   REWIND RRUNIT
   DVERT(INDV,IDP) = ARRAY(IPT + 1)
   IDLIM = DVERT(INDV,IDP)*2
   DO 1010 I=1,IDLIM
   INDV=INDV+1
   VERTEX(TOTVT2+1)=ARRAY(IPT+1+1)
1010 DVERT(INDV,IDP) = ARRAY(IPT + 1 + 1)
   INDV = INDV + 1
   TOTVT2=TOTVT2+IDLIM
   DO 1020 I=1,6
   DRECT(INDP,IDP)=FLDINF(I)
1020 INDP=INDP+1
   IDCNT(IDP)=IDCNT(IDP)+1
   IDUM=IDCNT(IDP)+1
   DVPNT(IDUM,IDP)=INDV
   GO TO 5
C  FINISHED WITH DO/DU FIELD PROCESSING
1030 CONTINUE
   IDP=0
   IDPP=ITRIG0+ITRIGU
   IF (ICK.LT.0) GO TO 20
   IF (ICK.EQ.0) GO TO 30
   IF (NOCL.GT.0) GO TO 6
   WRITE(6,800)
   CALL CMERP
6  CONTINUE
   NV=ARRAY(IPT+1)
   NVRT=NVRT+NV
   NOFLD=NOFLD+1
   NSAMP=(SAMEND-SAMSTR)/SAMINC+1
   FLDSAM=0
   IR=IPT+2
   NQ=NV-1
   NR=NQ-5
   IF (NR.GT.5) NQ=5
   IF (IR+NQ*2 - 1)
   * WRITE(6,600)NOFLD,ARRAY(IPT),SAMINC,LININC,
     * (LPRN,ARRAY(1),ARRAY(1+1),I=1H,IE,2)

```

RDD00800
 RDD00810
 RDD00820
 RDD00830
 RDD00840
 RDD00850
 RDD00860
 RDD00870
 RDD00880
 RDD00890
 RDD00900
 RDD00910
 RDD00920
 RDD00930
 RDD00940
 RDD00950
 RDD00960
 RDD00970
 RDD00980
 RDD00990
 RDD01000
 RDD01010
 RDD01020
 RDD01030
 RDD01040
 RDD01050
 RDD01060
 RDD01070
 RDD01080
 RDD01090
 RDD01100
 RDD01110
 RDD01120
 RDD01130
 RDD01140
 RDD01150
 RDD01160
 RDD01170
 RDD01180
 RDD01190
 RDD01200
 RDD01210
 RDD01220
 RDD01230
 RDD01240
 RDD01250
 RDD01260
 RDD01270
 RDD01280
 RDD01290
 RDD01300
 RDD01310
 RDD01320
 RDD01330
 RDD01340
 RDD01350
 RDD01360
 RDD01370
 RDD01380
 RDD01390
 RDD01400
 RDD01410
 RDD01420
 RDD01430
 RDD01440
 RDD01450
 RDD01460
 RDD01470
 RDD01480
 RDD01490
 RDD01500
 RDD01510
 RDD01520
 RDD01530
 RDD01540
 RDD01550
 RDD01560
 RDD01570
 RDD01580

FILE: RDATA

```

      IF (NR.LE.0) GO TO 7
      IR=IF+1
      IF=IR+NR*2 - 1
      WRITE (A.650) (LPRN,ARRAY(I),ARRAY(I+1),I=IR,IE,2)
7 CONTINUE
      IF (NSAMP*NOFEAT.GT.IDIM) GO TO 90
C*
C* POSITION TAPE FOR THIS FIELD
C*
      CALL FLDINT(FLDINF,FETVEC,NOFEAT)
      FLDSAM=0
      DO 10 LINE=LINSTR,LINEND,LININC
      LND(1)=0
      LND(2)=0
      IOBR = 2
      IDEE=1
      IF (IDPP.EQ.0) GO TO 1095
C BOTH DO AND DU TRIGGERS OFF --- SKIP AROUND
      DO 1040 IND=1,IDPP
      IOLIM=IND*IND
      DO 1050 I=1,IOLIM
      IDUM=(I-1)*6
      LOSTR=OJECT(IDUM+1,IND)
      LDEND=OJECT(IDUM+2,IND)
      LDINC=OJECT(IDUM+3,IND)
      DO 1040 II = LOSTR,LDEND,LDINC
      IF (II.NE.(LINE)) GO TO 1040
      LND(IND)=LND(IND)+1
      IDUM=LND(IND)
      LDOU(IDUM,IND) = I
1040 CONTINUE
1050 CONTINUE
1060 CONTINUE
      IF (LND(1).EQ.0.AND.LND(IDPP).EQ.0) GO TO 1095
C NO DO OR DU FOR THIS LINE
      IF (LND(1).GT.0) IOBR=1
      IF (IDPP.EQ.2.AND.LND(2).GT.0) IDEE=2
      DO 1090 IND=IOBR,IDEE
      IOLIM=LND(IND)
      IF (IOLIM.EQ.0) GO TO 1090
      IDUM=0
      DPINT(1,1) = 1
      DPINT(1,2) = 1
      DO 1080 I=1,IOLIM
      IOF=LDOU(I,IND)
      DVP=OVPNT(IOF,IND)
      CALL FDLINT(OVERT(DVP+1,IND),OVERT(DVP,IND),FL,LINE,SAMPS,NI)
      NPINT(I,IND)=NI
      IF (NI.EQ.0) GO TO 1080
      DO 1070 II=1,NI
1070 NPINT(II+IDUM,IND)=FL(II)
      IDUM=IDUM+NI
      DPINT(I+1,IND) = IDUM + 1
1080 CONTINUE
1090 CONTINUE
1095 CONTINUE
      CALL LINEPD(IDATA,ENDTAP)
      IF (ENDTAP.EQ.-1) GO TO 80
C*
C* FIND SAMPLE INTERSECTS FOR THIS LINE - NI=NO. OF INTERSECTS
C*
      CALL FDLINT(ARRAY(IPT+2),NV,FL,LINE,SAMPS,NI)
C*
C* STORE DATA ON THIS LINE INTO OUTPUT BUFFER
C*
      REAL RWRD
      EQUIVALENCE (RWRD,ITGWRD)
      MODSS=MOD(SAMSTR,SAMINC)
      DO 60 I=1,NI,2
      IR=(FL(I)-SAMSTR)/SAMINC+1
      IF=(FL(I+1)-SAMSTR)/SAMINC+1
      IF (MODSS.NE.MOD(FL(I),SAMINC)) IR=IR+1
      IF (IR.GT.IE) GOTO 60
      IF (IDPP.EQ.0) GOTO 2055
      IF (LND(IDPP).EQ.0.AND.LND(IDEE).EQ.0) GOTO 2055
      DO 2050 IND=IOBR,IDEE
      IOLIM=LND(IND)
      IF (IOLIM.EQ.0) GOTO 2050
      IOSIT=1

```

R0001590
R0001600
R0001610
R0001620
R0001630
R0001640
R0001650
R0001660
R0001670
R0001680
R0001690
R0001700
R0001710
R0001720
R0001730
R0001740
R0001750
R0001760
R0001770
R0001780
R0001790
R0001800
R0001810
R0001820
R0001830
R0001840
R0001850
R0001860
R0001870
R0001880
R0001890
R0001900
R0001910
R0001920
R0001930
R0001940
R0001950
R0001960
R0001970
R0001980
R0001990
R0002000
R0002010
R0002020
R0002030
R0002040
R0002050
R0002060
R0002070
R0002080
R0002090
R0002100
R0002110
R0002120
R0002130
R0002140
R0002150
R0002160
R0002170
R0002180
R0002190
R0002200
R0002210
R0002220
R0002230
R0002240
R0002250
R0002260
R0002270
R0002280
R0002290
R0002300
R0002310
R0002320
R0002330
R0002340
R0002350
R0002360
R0002370

FILE: R00DATA

```

MEAND0=MEAND0
IF (IDRR.EQ.IDFE) GOTO 2003
IF (IND.EQ.2) IDSIT=2
IF (IND.EQ.2) MEAND0=MEANDU
GOTO 2009
2003 IF (IDRR.EQ.1.AND.IDPP.EQ.2) GOTO 2009
IF (ITRIGU.EQ.0) GOTO 2009
IDSIT=2
MEAND0=MEANDU
2009 CONTINUE
DO 2040 K=1.IDLIM
NDIN=NDINT(K.IND)
IF (NDIN.EQ.0) GOTO 2040
OPIN=OPINT(K.IND)
DO 2010 KK=1.NDIN
2010 DIN(KK)=DINT(OPIN+KK-1.IND)
IDUM=0
DO 2020 KK=1.NDIN+2
IDUM=IDUM+1
IDB(IDUM)=(DIN(KK)-SAMSTR)/SAMINC+1
IDE(IDUM)=(DIN(KK+1)-SAMSTR)/SAMINC+1
IF (MODSS.ME.MOD(DIN(KK).SAMINC)) IDB(IDUM)=IDB(IDUM)+1
2020 CONTINUE
DO 2030 KK=1.IDUM
IDS=IDB(KK)
IDF=IDF(KK)
IF (IDS.GT.IE.OP.IH.GT.IDF) GOTO 2030
IF (IDS.LE.IH) IDS=IH
IF (IDF.GT.IE) IDF=IE
IF (IDS.GT.IDF) GOTO 2030
DO 2025 KKK=IDS.IDF
NDIU(IND)=NDIU(IND)+1
DO 2023 KKKK=1.NOFEAT
DUMMY1=KKK+NSAMP*(KKKK-1)
2023 IDATA(DUMMY1)=MEAND0
2025 CONTINUE
2030 CONTINUE
2040 CONTINUE
IF (IDSIT.EQ.1.AND.NDIU(IND).GT.0) DOFLAG=1
IF (IDSIT.EQ.2.AND.NDIU(IND).GT.0) DUFLAG=1
2050 CONTINUE
2055 DODU=DOFLAG+DUFLAG
DO 50 J=1H.IF
FLDSAM=FLDSAM+1
DO 50 K=1.NOFEAT
IWRD=IWRD+1
DUMMY2=J+NSAMP*(K-1)
RWRD=IDATA(DUMMY2)
DUMMY3=IF(INDX-1+IWRD+BUFSIZ*(IBUF-1)
APRAY(DUMMY3)=ITGVRD
IF (IWRD.LT.BUFSIZ) GOTO 50
TOTW=IOT+RWRD+IWRD
IF (TOTW.GT.MADS) GOTO 35
DUMMY4=IF(INDX+BUFSIZ*(IBUF-1)
CALL RWRITE(ADDRS.ARRAY(DUMMY4).BUFSIZ,LSTAT(IBUF))
IBUF=IBUF+1
ADDRS=ADDRS+BUFSIZ
IF (IBUF.GT.IBUFS) IBUF=1
40 IF (LSTAT(IBUF).EQ.1) GOTO 40
IWR=0
50 CONTINUE
60 CONTINUE
10 CONTINUE
IPT = IPT + NV*2 + 2
APRAY(IPT)=FLDSAM
01 15 I=1.6
IPT=IPT+1
15 ARRAY(IPT)=FLDINF(I)
IPT=IPT+1
IF (IPT+30 .GT. RESERV) GO TO 70
GO TO 5

C*
C* CLASS NAME CARD ENCOUNTERED - RERFAD PREVIOUS CARD TO GET NAME
C*
20 NOCL=NOCL+1
IF (NOCL.GT.1) GO TO 25
READ(30.100)NATCLS
REWIND 30
GO TO 4

```

R0002380
R0002390
R0002400
R0002410
R0002420
R0002430
R0002440
R0002450
R0002460
R0002470
R0002480
R0002490
R0002500
R0002510
R0002520
R0002530
R0002540
R0002550
R0002560
R0002570
R0002580
R0002590
R0002600
R0002610
R0002620
R0002630
R0002640
R0002650
R0002660
R0002670
R0002680
R0002690
R0002700
R0002710
R0002720
R0002730
R0002740
R0002750
R0002760
R0002770
R0002780
R0002790
R0002800
R0002810
R0002820
R0002830
R0002840
R0002850
R0002860
R0002870
R0002880
R0002890
R0002900
R0002910
R0002920
R0002930
R0002940
R0002950
R0002960
R0002970
R0002980
R0002990
R0003000
R0003010
R0003020
R0003030
R0003040
R0003050
R0003060
R0003070
R0003080
R0003090
R0003100
R0003110
R0003120
R0003130
R0003140
R0003150
R0003160

```

25 CLSNAM=NXTCLS  
WFAD(30,100)NXTCLS  
WFWIND 30  
GO TO 31  
C* EMPTY LAST BUFFER AND RETURN TO PROCESS DATA FOR THIS CLASS.  
30 CLSNAM=NXTCLS  
LAST=1  
C*  
31 TOTWRD=TOTWRD+IWRD  
IF(TOTWRD.GT.NWDS) GOTO 35  
DUMMY=WFINDEX+BUFSIZ*(IBUF-1)  
CALL WRITE(AADDRESS,ARRAY(DUMMY),IWRD,LSTAT(IBUF))  
TOTPTS=TOTPTS+IWRD/WOFEAT  
IF(TOTWRD.TOTPTS.LE.NWDS)RETURN  
WRITE(A,200)NWDS  
RETURN  
35 WRITE(A,200)NWDS  
CALL CMERP  
70 WRITE(A,300)RESERV  
CALL CMERP  
80 WRITE(A,400)  
CALL CMERP  
90 WRITE(A,700)IDIM  
CALL CMERP  
100 FORMAT(10X,A4)  
140  
200 FORMAT('///' DESIGNATED OTHER OR UNIDENTIFIABLE FIELDS INPUT')  
200 FORMAT(' TOO MUCH DATA REQUESTED--PIXELS*(CHANNELS+1) CANNOT EXCEE  
*('),10)  
300 FORMAT(' STORAGE REQUIRED FOR FIELD DEFINITION INFORMATION EXCEEDS  
* THE DIMENSION LIMIT OF(.15)  
400 FORMAT(' END-OF-TAPE REACHED BEFORE END OF FIELD')  
500 FORMAT('//40X,'FIELDS TO BE CLUSTERED FOR CLASS'.1X,A4//  
* T36.'SAMPLE'.T45.'LINE'/T20.'FIELD NAME'.T36.'INC.'.  
* T45.'INC.'.T73.'VERTICES (SAMPLE.LINE)'/)  
600 FORMAT(1X,T16,13,T22,A4,T36,14,T45,14,T60,  
* S(A).14.,.,.14.)'.1X))  
650 FORMAT(1X,T60,5(A).14.,.,.14.)'.1X))  
700 FORMAT(' NO OF PIXELS TO BE UNPACKED PER SCAN EXCEEDS THE DIMENSI  
*ON LIMIT OF(.15)  
800 FORMAT('// INPUT ERROR - A CLASSNAME CAPD MUST BE INPUT BEFORE A GRP  
*ROUP OF FIELDS')  
RETURN  
END
```

10. SELECT PROCESSOR

FILE SELECT

```

C      SUBROUTINE SELECT(ARRAY, TOP)
C      IMPLICIT INTEGER (A-H, O-Z)
C-----
C      CALL..    CALL SELECT(ARRAY, TOP)
C      ARGS..    ARRAY = SFE 'MONTOR'
C               TOP   = SFE 'MONTOR'
C
C      PURPOSE.. COORDINATES THE VARIOUS ROUTINES
C               FOR 'FEATURE SELECTION' STEP
C
C      RETURNS.. NONE
C-----
C
C      INCLUDE COMBK1, LIST
C-----
C      INCLUDE COMRK6, LIST
C      DIMENSION ARRAY(1), SUBRAY(12000)
C      DATA SUBSIZ/12000/
C      * THE ARRAY SUBRAY IS USED IN SELECT FOR VARIABLE DIMENSIONING
C      * INCLUDE COMRK7, LIST
C      * COMMON/INFORM/NOCLS2, NOSUB2, NOFET2, VARSZ2, TOTVT2, NOFLD2,
C      *      AVAR2, COVAR2, CLSID2, SUBNO2, SUBDS2, FLDSV2, VERTX2,
C      *      FETVC2(30), SUBVC2(75), SUBPTR(75), CLSVC2(60),
C      *      KEPPTS(60), NOGRP, GRPNAM(60), GRPDEX(61),
C      *      GRPCHK(61), GROUPS(124)
C      * COMMON/GLOBAL/HEAD(63), MAPTAP, DATAPE, SAVTAP, BMFILE, BMKEY,
C      *      HISFIL, HISKEY, TRFORM, ERIPTR, ERPKEY, MAPUNT, NOFILE,
C      *      DRUMAD, DRMWDS, PAGSIZ, DATFIL, STAFIL, ASAV, ASAVFL
C      *      NHSTUN, NHSTFI, SCTRUN, MAPFIL
C      *      DOTUNT, DOTFIL, NCHPAS, TRNSFL, BMTRFL, HISTFL, PCHUNT,
C      *      CRDUNT, PRUNT, RANDIO
C      * COMMON/FSL/CFAC, TOTMSR, SEPMSR, PRCKEY, CRIKEY, INCFET,
C      *      INCVEC(30), ICOUNT, SETWGT, EVALBF(100), FETVC4(30)
C      *      ,NOFET4, VARSZ4, COMBAS, DTAB4, WGH514, RESTVC(10), DIVSIZ
C      *      ,STATKY, ADRESO, ADRESP, ADRESF, ADRSH1, ADRSH2
C      * INTEGER ADRESO, ADRESP, ADRESF, ADRSH1, ADRSH2, STATKY
C      * DOUBLE PRECISION CFAC, TOTMSR, SEPMSR
C
C      CSEND
C
C      COMMON BLOCK FSL IS USED ONLY BY THE 'SELECT' PROCESSOR
C
C      DEFINITIONS
C      PRCKEY - KEY INDICATING WHICH PROCEDURE TO EXECUTE
C              1 - EXHAUSTIVE SEARCH
C              2 - WITHOUT REPLACEMENT
C              3 - DAVIDON
C              4 - EVALUATE A USER INPUT B-MATRIX
C              5 - EVALUATE SPECIFIC CHANNELS INPUT BY USER
C      CRIKEY - KEY INDICATING WHICH CRITERIA IS TO BE USED
C              FOR MEASURING SEPARABILITY.
C              1 - WEIGHTED AV. DIVERGENCE
C              2 - WEIGHTED AV. TRANSFORMED DIVERGENCE
C              3 - WEIGHTED AV. KHATTACHARYYA DISTANCE
C      INCFET - NO. OF CHANNELS TO INCLUDE IN THE 'BEST' SET. THIS IS
C              USER INPUT ON THE 'INCLUDE' CONTROL CARD.
C      INCVEC - VECTOR CONTAINING THE CHANNELS TO BE INCLUDED. INPUT
C              ON THE 'INCLUDE' CARD. MEANINGFUL ONLY IF WITHOUT
C              REPLACEMENT PROCEDURE IS EXECUTED.
C      ICOUNT - MAX. NO. OF ITERATIONS IN DAVIDON PROCEDURE.

```

SEL00010
 SEL00020
 SEL00030
 SEL00040
 SEL00050
 SEL00060
 SEL00070
 SEL00080
 SEL00090
 SEL00100
 SEL00110
 SEL00120
 SEL00130
 SEL00140
 SEL00150
 SEL00160
 SEL00170
 SEL00180
 SEL00190
 SEL00200
 SEL00210
 SEL00220
 SEL00230
 SEL00240
 SEL00250
 SEL00260
 SEL00270
 SEL00280
 SEL00290
 SEL00300
 SEL00310
 SEL00320
 SEL00330
 SEL00340
 SEL00350
 SEL00360
 SEL00370
 SEL00380
 SEL00390
 SEL00400
 SEL00410
 SEL00420
 SEL00430
 SEL00440
 SEL00450
 SEL00460
 SEL00470
 SEL00480
 SEL00490
 SEL00500
 SEL00510
 SEL00520
 SEL00530
 SEL00540
 SEL00550
 SEL00560
 SEL00570
 SEL00580
 SEL00590
 SEL00600
 SEL00610
 SEL00620
 SEL00630
 SEL00640
 SEL00650
 SEL00660
 SEL00670
 SEL00680
 SEL00690
 SEL00700
 SEL00710
 SEL00720
 SEL00730
 SEL00740
 SEL00750
 SEL00760

FILE SELECT

```

C*      SETWGT - TRIGGER INDICATING WHETHER OR NOT DEFAULT WEIGHTS      SEL00770
C*      ARE TO BE SET.                                                  SEL00780
C*      EVALBF - BUFFER CONTAINING ALL USER 'EVALUATE' REQUESTS.        SEL00790
C*      EVALRF(1) = NO. OF CHANNELS FOR FIRST REQUEST                  SEL00800
C*      (2-N) = A SET OF CHANNELS TO BE EVALUATED                     SEL00810
C*      (N+1) = NO. OF CHANNELS ON SECOND REQUEST                      SEL00820
C*      -                                                                SEL00830
IEY0331 COMMENTS DELETED *****
COMMON/RESTKN/ KPPPTS(60), IPRIOR, KBEST, NCPASS                      SEL01190
DIMENSION ILABLX(13), ILARLY(13)                                     SEL01200
DIMENSION FETVEC(30)                                                 SEL01210
DIMENSION PER(6), INOPER(6), PERM(248)                               SEL01220
DATA INOPER/1,13,58,138,213,249/                                     SEL01230
DATA PERM/1,2,1,3,1,4,1,5,1,6,1,7,                                SEL01240
* 1,2,3,1,2,4,1,2,5,1,2,6,1,2,7,1,3,4,1,3,5,1,3,6,1,3,7,          SEL01250
* 1,4,5,1,4,6,1,4,7,1,5,6,1,5,7,1,6,7,                             SEL01260
* 1,2,3,4,1,2,3,5,1,2,3,6,1,2,3,7,1,2,4,5,1,2,4,6,              SEL01270
* 1,2,4,7,1,2,5,6,1,2,5,7,1,2,6,7,1,3,4,5,1,3,4,6,              SEL01280
* 1,3,4,7,1,3,5,6,1,3,5,7,1,3,6,7,1,4,5,6,                       SEL01290
* 1,4,5,7,1,4,6,7,1,5,6,7,                                         SEL01300
* 1,2,3,4,5,1,2,3,4,6,1,2,3,4,7,1,2,3,5,6,1,2,3,5,7,          SEL01310
* 1,2,3,6,7,1,2,4,5,6,1,2,4,5,7,1,2,4,6,7,1,2,5,6,7,          SEL01320
* 1,3,4,5,6,1,3,4,5,7,1,3,4,6,7,1,3,5,6,7,1,4,5,6,7,          SEL01330
* 1,2,3,4,5,6,1,2,3,4,5,7,1,2,3,4,6,7,1,2,3,5,6,7,          SEL01340
* 1,2,4,5,6,7,1,3,4,5,6,7/                                         SEL01350
DIMENSION IPSCHK(8)                                                  SEL01360
DIMENSION FETSAV(30), FTSAB(30)                                     SEL01370
DOUBLE PRECISION SMALL                                              SEL01380
SMALL = 2**35                                                       SEL01390
JTIME = 1                                                            SEL01400
NFSAVE = 0                                                            SEL01410
10 CONTINUE                                                           SEL01420
CALL SETUP4 (ARRAY, TOP, STOPFG, JTIME, SUBRAY, SUBSIZ)              SEL01430
IF (STOPFG.NE.0) GO TO 9                                             SEL01440
IF (PRCKEY.NE.6) GO TO 9                                             SEL01450
DO 6 I=1,8                                                           SEL01460
6 IPSCHK(I) = 0                                                       SEL01470
DO 7 I=1,NOFET2                                                       SEL01480
7 K=(FETVC2(I)-1)/NCPASS + 1                                         SEL01490
IPSCHK(K) = 1                                                         SEL01500
NPASS = 0                                                             SEL01510
DO 8 I=1,8                                                           SEL01520
8 NPASS = NPASS + IPSCHK(I)                                           SEL01530
NFPPS = NOFET2/NPASS                                                 SEL01540
IDUM = NFPPS*NPASS                                                   SEL01550
IF (NOFET2.NE.IDUM) GO TO 90                                          SEL01560
IF (KREST.LT.2.OR.KREST.GE.NPASS) GO TO 90                          SEL01570
NOFET4 = NFPPS*KREST                                                 SEL01580
NFSAVE = NOFET4                                                       SEL01590
KPASS = KBEST                                                         SEL01600
9 CONTINUE                                                            SEL01610
JTIME = JTIME+1                                                       SEL01620
IF (STOPFG.EQ.0) GO TO 5                                             SEL01630
C* SET FETVC2 FOR CLASSIFY                                           SEL01640
IF (NFSAVE.NE.0) NOFET4=NFSAVE                                       SEL01650
DO 4 I=1,NOFET4                                                       SEL01660
4 FETVC2(I)=FETVC4(I)                                                 SEL01670
CALL ORDER(FETVC2,NOFET4)                                             SEL01680
NOFET2 = NOFET4                                                       SEL01690
RETURN                                                                SEL01700
5 CONTINUE                                                            SEL01710
C* SET ADRESO FOR RANDOM ACCESS DRUM FILE                             SEL01720
C* ADRESO=DRUMAD                                                      SEL01730
C* JREST = 0                                                           SEL01740
C* PRELIM--TAKE CARE OF PRELIMINARIES                                SEL01750
C* COMPUTE SEPARABILITY MEASURE AND INTERCLASS MEASURES USING ALL    SEL01760
C* FEATURES, AND SAVE ON SCRATCH FILE FOR LATER PRINTING. IF CRIKEY=1 SEL01770
C* SET DEFAULT WEIGHTS IF WEIGHTS NOT INPUT, ALSO COMPUTE 'S' MATRIX. SEL01780
C*                                                                      SEL01790
SRASE=1                                                                SEL01800
S1=SRASE                                                              SEL01810
SHASF=S1+NOCLS2*VARSZ2+1                                             SEL01820
IF (CRIKEY.NE.1) SHASF=1                                             SEL01830
SLEFT=SUBSIZ-SRASE                                                   SEL01840
                                                                    SEL01850
                                                                    SEL01860

```


FILE SELECT

```

      CALL PRELIM (ARRAY (COVAR2), ARRAY (AVAR2), ARRAY (DTAB4),
      *   ARRAY (WGHS14), SUBRAY (S1), SUBRAY (SHASE), SLEFT)
      IF (PRCKEY .EQ. 5) GO TO 60
      IF (PRCKEY .EQ. 4) GO TO 12
      IF (PRCKEY .EQ. 6) GO TO 15
11  JRFST=JBEST+1
      NFSAVE=NOFET4
      NOFET4=BESTVC (JBEST)
      IF (NOFET4 .LE. 0) GO TO 60
C*
C*   IF DAVIDON PROCEDURE INDICATED, FIND BEST SET OF FEATURES BY
C*   WITHOUT REPLACEMENT, IF FIRST GUESS B-MATRIX WAS NOT INPUT.
C*
12 IF (PRCKEY .NE. 3) GO TO 15
C*
C*   SET ADDRESSES FOR RANDOM ACCESS DRUM FILE
C*
      ADRESF=ADRES0+DIVSIZ*2
      ADRESF=ADRESF+NOFET4*NOFET2*2
      ADRESH1=ADRESF+NOFET4*NOFET2*2
      ADRESH2=ADRESH1+(NOFET4*NOFET2*2)**2
C*
C*   WAS FIRST GUESS B-MATRIX INPUT
C*
      IF (HMKEY .EQ. 1) GO TO 15
      SAVPRC=3
      PRCKEY=2
C*
C*   COMPUTE BASES FOR ARRAYS OF 'BEST' SET OF FEATURES
C*   TRANSFORMED COVARIANCES AND MEANS STORED IN DOUBLE PRECISION
C*
15 VARSZ4=NOFET4*(NOFET4+1)/2
      COVAR4=CORRAS
      AVAR4=COVAR4 + NOCLS2*VARSZ4*2
      CORR55 = AVAR4 + NOCLS2 * NOFET4 *2
      IF (CORR55 .LE. TOP) GO TO 20
      WRITE (6,200) CORR55
      CALL CMERR
C*
C*   SUBRAY STORAGE - STORE 'S' ARRAYS ONLY IF CRIKEY=1, STORE PARTIALS
C*   ONLY IF PRCKEY=3, STORE B-MATRIX IF PRCKEY=3 OR 4.
C*
20 S2=S1 + NOCLS2*VARSZ2
      R1=S2 + NOCLS2*VARSZ4*2
      IF (CRIKEY .NE. 1) H1=1
      P1=H1 + NOFET4*NOFET2*2
      SHASE=P1 + NOFET4*NOFET2*2
      IF (PRCKEY .NE. 3) SHASE=P1
      IF (PRCKEY .LT. 3) SHASE=B1
      SLEFT=SUHSIZ-SHASE
      IF (SHASE .LE. SUHSIZ) GO TO 25
      WRITE (6,100) SHASE
      CALL CMERR
C*
C*   PERFORM THE OPTIMIZATION PROCEDURE INDICATED BY PRCKEY
C*
25 GO TO (30,35,40,45,85,87),PRCKEY
C*
C*   EXHAUSTIVE SEARCH PROCEDURE
C*
30 CALL EXSRCH (ARRAY (COVAR2), ARRAY (AVAR2), ARRAY (DTAB4), ARRAY (WGHS14),
      *   ARRAY (COVAR4), ARRAY (AVAR4), SUBRAY (S1), SUBRAY (S2),
      *   SUBRAY (SHASE), SLEFT)
      GO TO 50
C*
C*   WITHOUT REPLACEMENT PROCEDURE
C*
35 CALL WHRPLC (ARRAY (COVAR2), ARRAY (AVAR2), ARRAY (DTAB4), ARRAY (WGHS14),
      *   ARRAY (COVAR4), ARRAY (AVAR4), SUBRAY (S1), SUBRAY (S2),
      *   SUBRAY (SHASE), SLEFT)
      GO TO 50
C*
C*   DAVIDON PROCEDURE
C*
40 CALL DAVIDN (ARRAY (COVAR2), ARRAY (AVAR2), ARRAY (DTAB4), ARRAY (WGHS14),
      *   ARRAY (COVAR4), ARRAY (AVAR4), SUBRAY (S1), SUBRAY (S2))

```

```

SEL01870
SEL01880
SEL01890
SEL01900
SEL01910
SEL01920
SEL01930
SEL01940
SEL01950
SEL01960
SEL01970
SEL01980
SEL01990
SEL02000
SEL02010
SEL02020
SEL02030
SEL02040
SEL02050
SEL02060
SEL02070
SEL02080
SEL02090
SEL02100
SEL02110
SEL02120
SEL02130
SEL02140
SEL02150
SEL02160
SEL02170
SEL02180
SEL02190
SEL02200
SEL02210
SEL02220
SEL02230
SEL02240
SEL02250
SEL02260
SEL02270
SEL02280
SEL02290
SEL02300
SEL02310
SEL02320
SEL02330
SEL02340
SEL02350
SEL02360
SEL02370
SEL02380
SEL02390
SEL02400
SEL02410
SEL02420
SEL02430
SEL02440
SEL02450
SEL02460
SEL02470
SEL02480
SEL02490
SEL02500
SEL02510
SEL02520
SEL02530
SEL02540
SEL02550
SEL02560
SEL02570
SEL02580
SEL02590
SEL02600
SEL02610
SEL02620

```

FILE SELECT

```

      *      ,SURRAY(B1),SUBRAY(P1),SUBRAY(SBASE),SLEFT)
      CALL WRBTMT(SURRAY(H1),NOFET4,NOFET2,FETVC2)
      GO TO 50
C*
C*      USER INPUT B-MATRIX
45  CALL USERIN(ARRAY(COVAR2),ARRAY(AVAR2),ARRAY(DTAB4),ARRAY(WGHS14),
      *      ,ARRAY(COVAR4),ARRAY(AVAR4),SUBRAY(S1),SUBRAY(S2)
      *      ,SURRAY(B1),SUBRAY(SBASE),SLEFT)
C*
C*      GENERATE REPORTS
C*
50  CALL GENRPT(ARRAY(CLSID2),ARRAY(WGHS14),ARRAY(DTAB4),
      *      ,SURRAY(SBASE),SLEFT,FETVEC)
      CALL PLOT(SURRAY(SBASE),ARRAY(DTAB4),DIVSIZ,MAXX,ILABLX,ILABLY,
      *      ,ICODE,IOPT)
C*
      IF (SAVPHC.NE.3) GO TO 11
      SAVPRC=0
      PRCKEY=3
      GO TO 20
C*
C*      PERFORM EVALUATE REQUEST
C*
60  IV=1
      ISAVE=PRCKEY
      PRCKEY=5
70  NOFET4=EVALHF(IV)
      IF (NOFET4.GT.0) GO TO 75
      PRCKEY=ISAVE
      GO TO 10
75  DO 90 I=1,NOFET4
      IV=IV+1
80  FETVEC(I) = EVALHF(IV)
C
C      RENUMBERING CHANNELS IN REFERENCE TO SUBSET OF CHANNELS
C
      DO 82 I=1,NOFET2
      DO 82 J=1,NOFET4
      IF (FETVEC(J).NE.FETVC2(I)) GO TO 82
      FETVC4(J) = I
82  CONTINUE
      CALL ORDER(FETVC4,NOFET4)
C*
C*      GO COMPUTE BASE ADDRESSES FOR REDUCED ARRAYS
C*
      GO TO 15
85  CONTINUE
      CALL EVLFET(ARRAY(COVAR2),ARRAY(AVAR2),ARRAY(DTAB4),ARRAY(WGHS14),
      *      ,ARRAY(COVAR4),ARRAY(AVAR4),SUBRAY(S1),SUBRAY(S2)
      *      ,SURRAY(SBASE),SLEFT)
      CALL GENRPT(ARRAY(CLSID2),ARRAY(WGHS14),ARRAY(DTAB4),
      *      ,SURRAY(SBASE),SLEFT,FETVEC)
      CALL PLOT(SURRAY(SBASE),ARRAY(DTAB4),DIVSIZ,MAXX,ILABLX,ILABLY,
      *      ,ICODE,IOPT)
      IV=IV+1
      GO TO 70
C
87  BEST K OF N PASSES
91  CONTINUE
      NCNT = NPASS + 1 - KPASS
      IDUM = INDPER(KPASS - 1)
      NCNTP = (INDPER(KPASS) - IDUM)/KPASS
      DO 99 I=1,NCNT
      DO 98 II=1,NCNTP
      DO 92 III=1,KPASS
      IDM=IDUM*(II-1)*KPASS+III-1
      PER(III) = PERM(IDM) + III-1
      IF (PER(III).GT.NPASS) GO TO 97
92  CONTINUE
      DO 94 III=1,KPASS
      IDM=PER(III)
      DO 93 II=1,NFPPS
      IDMA = NFPPS*(III-1) + II
      IDMB = NFPPS*(IDM-1) + II
      FETVC4(IDMA) = IDM
      FETVC4(IDMB) = FETVC2(IDMB)
93  CONTINUE
94

```

```

SEL02630
SEL02640
SEL02650
SEL02660
SEL02670
SEL02680
SEL02690
SEL02700
SEL02710
SEL02720
SEL02730
SEL02740
SEL02750
SEL02760
SEL02770
SEL02780
SEL02790
SEL02800
SEL02810
SEL02820
SEL02830
SEL02840
SEL02850
SEL02860
SEL02870
SEL02880
SEL02890
SEL02900
SEL02910
SEL02920
SEL02930
SEL02940
SEL02950
SEL02960
SEL02970
SEL02980
SEL02990
SEL03000
SEL03010
SEL03020
SEL03030
SEL03040
SEL03050
SEL03060
SEL03070
SEL03080
SEL03090
SEL03100
SEL03110
SEL03120
SEL03130
SEL03140
SEL03150
SEL03160
SEL03170
SEL03180
SEL03190
SEL03200
SEL03210
SEL03220
SEL03230
SEL03240
SEL03250
SEL03260
SEL03270
SEL03280
SEL03290
SEL03300
SEL03310
SEL03320
SEL03330
SEL03340
SEL03350
SEL03360
SEL03370
SEL03380

```

FILE SELECT

CALL EVLFET (ARRAY (COVAR2), ARRAY (AVAR2),	SEL03390
* ARRAY (DTAH4), ARRAY (WGHS14), ARRAY (COVAR4), ARRAY (AVAR4),	SEL03400
* SUBRAY (S1), SUBRAY (S2), SUBRAY (SBASE), SLEFT)	SEL03410
IF (SMALL.LT.SEPMSR) GO TO 96	SEL03420
DO 95 III = 1, NOFET4	SEL03430
FISAV (III) = FETVEC (III)	SEL03440
95 FETSAV (III) = FETVC4 (III)	SEL03450
SMALL = SEPMSR	SEL03460
96 CONTINUE	SEL03470
CALL GENRPT (ARRAY (CLS102), ARRAY (WGHS14),	SEL03480
* ARRAY (DTAR4), SUBRAY (SBASE), SLEFT, FETVEC)	SEL03490
97 CONTINUE	SEL03500
98 CONTINUE	SEL03510
99 CONTINUE	SEL03520
DO 101 I = 1, NOFET4	SEL03530
FETVEC (I) = FISAV (I)	SEL03540
101 FETVC4 (I) = FISAV (I)	SEL03550
WRITE (6, 1010)	SEL03560
1010 FORMAT (1H1, 'BEST SEPARABILITY MEASURE', //)	SEL03570
WRITE (6, 1000) SMALL	SEL03580
WRITE (6, 1020)	SEL03590
1020 FORMAT (//, ' CORRESPONDING FEATURES', //)	SEL03600
WRITE (6, 1000) (FETVEC (I), I = 1, NOFET4)	SEL03610
1000 FORMAT ()	SEL03620
GO TO 10	SEL03630
90 WRITE (6, 250) NOFET4, (FETVC2 (I), I = 1, NOFET2)	SEL03640
250 FORMAT (' ERROR IN INPUT CHANNELS', //, 1X, I2, 30 (1X, I2))	SEL03650
CALL CMERR	SEL03660
100 FORMAT (' CORE OVERFLOW IN SUBRAY-', I6, ' STORAGE LOCATIONS NECESSARY	SEL03670
*Y FOR THIS PROBLEM')	SEL03680
200 FORMAT (' CORE OVERFLOW IN ARRAY-', I6, ' STORAGE LOCATIONS NEEDED FOR	SEL03690
*R THIS PROBLEM')	SEL03700
END	SEL03710

[illegible]

FILE: AVEDIV

```

      DO 25 K=1, KK
      L=K-1
25  WRKRY(IW1+L)=WRKRY(IW2+L)-WRKRY(IW)+L
      CALL MT2(WRKRY(ICV),WRKRY(IW1),WRKRY(IW2),NOFET4,NOFET2)
      DO 26 K=1, KK
      L=K-1
26  PARTLS(K)=WRKRY(IW2+L)*PARTLS(K)
30  CONTINUE
      SMSR = -(CFAC*SMSR/2. - NF)
      IF (IPART.LT.0) RETURN
      DO 40 K=1, KK
40  PARTLS(K)=-CFAC*PARTLS(K)
      RETURN
100 FORMAT(' REDUCED COVARIANCE MATRIX FOR CLASS',I3,' IS NOT POSITIVE
      * DEFINITE')
200 FORMAT(' MORE STORAGE NEEDED IN SUB. AVEDIV FOR WORK ARRAY--WORK
      * IZF=',I7)
      END
```

AVE00810
AVE00820
AVE00830
AVE00840
AVE00850
AVE00860
AVE00870
AVE00880
AVE00890
AVE00900
AVE00910
AVE00920
AVE00930
AVE00940
AVE00950
AVE00960
AVE00970
AVE00980

FILE: RHTCHH

```

      SUBROUTINE RHTCHH(SMSR,COVMTX,AVEMTX,WEIGHT,DIVTAB,
      COVM12,AVEM12,
      WRKRY,IWRKSZ,IPART,PARTLS,RMAT,IFULL)
C      INCLUDE COMMR7.LIST
C      INTEGER VARSZ4,DIVSZ,VARSZ2
C      INCLUDE COMMR1.LIST
      COMMON/INFORM/NOCLS2,NOSUR2,NOFET2,VARSZ2,TOTVT2,NOFLD2,
      AVAR2,COVAR2,CLS1D2,SUHN2,SURDS2,FLDSV2,VERTX2,
      FETVC2(30),SURVC2(75),SURPTR(75),CLSV2(60),
      KPPTS(60),NOGRP,GRPNAM(60),GRPDEX(61),
      GRPCHK(41),GROUPS(124)
      COMMON/FSL/CFAC,TOTMSR,SEPMSR,PRCKEY,CRIKEY,INCFET,
      INCVEC(30),ICOUNT,SEIWTG,EVALHF(100),FETVC4(30),
      NOFET4,VARZ4,CORHAS,DIHA4,WGH514,RESTVC(10),DIVSZ
      ,STATKY,ADRFSD,AURESP,ADRESF,ADPSH1,ADPSH2
      INTEGER ADHFSO,ADRESF,ADHFSF,ADHSH1,ADHSH2,STATKY
      DOUBLE PRECISION CFAC,TOTMSR,SEPMSR
C$END
C*
C*      SUBROUTINE TO COMPUTE THE INTERCLASS BHATTACHARYA DISTANCE,
C*      THE WEIGHTED AVERAGE DISTANCE, AND THE PARTIALS WITH RESPECT
C*      TO R.
C*
C*      IF IFULL=1 COMPUTE H. DISTANCE FOR ALL 'NOFET' CHANNELS.
C*      PARTIALS CANNOT BE COMPUTED WHEN IFULL=1.
C*
      DOUBLE PRECISION DIVTAB(DIVSZ),DET1,DET2,DET3
      DOUBLE PRECISION SMSR
      DOUBLE PRECISION RMAT,PARTLS(1)
      DOUBLE PRECISION COVM2,AVEM2,WRKRY(1),T(30),RNUM
      DIMENSION COVMTX(VARSZ2,NOCLS2), COVM2(VARSZ4,NOCLS2),
      AVEMTX(NOFET2,NOCLS2), AVEM2(NOFET4,NOCLS2),
      WEIGHT(DIVSZ),RMAT(1)
      IVSZ=VARSZ4
      NF=NOFET4
      IF(IFULL.EQ.1) IVSZ=VARSZ2
      IF(IFULL.EQ.1) NF=NOFET2
      ICV1=1
      ICV2=ICV1+IVSZ
      IW1=ICV2+IVSZ
      IW2=IW1+IVSZ
      ITFST=IW2+NF
      IF(IPART.LT.0) GO TO 3
      ZERO PARTIALS
      IQ=NOFET2*NOFET4
      DO 2 IK=1,IQ
      2 PARTLS(IK)=0.0
      IW3=IW2+NOFET2
      IW4=IW3+NOFET2
      IW5=IW4+MAX0(VARSZ2,IQ)
      IW6=IW5+IQ
      IW7=IW6+IQ
      ITFST=IW7+IQ
      3 CONTINUE
      IF(IWRKSZ/2.GE.ITFST) GO TO 1
      WRITE(6,300) IWRKSZ
      CALL CMERR
      1 CONTINUE
      SMSR=0
      NM=0
      IC=NOCLS2-1
      DO 60 I=1,IC
      C*      FIND INVERSE AND DETERMINANT FOR CLASS I
      DO 5 IK=1,IVSZ
      IF(IFULL.EQ.1) WRKRY(IK)=(COVMTX(IK,I)
      IF(IFULL.EQ.1) WRKRY(IK)=COVM2(IK,I)
      5 CONTINUE
      CALL COLIN(VWRKRY(ICV1),NF,IFRR,3,DET1)
      DET1=1./DET1
      IF(IFRR.EQ.0) GO TO 6
      WRITE(6,100) I
      GO TO 60
      6 IM=I+1
      DO 50 J=IM,NOCLS2
      NM=NM+1
      C*      COMPUTE INVERSE AND DETERMINANT FOR CLASS J - AND -
      C*      COMPUTE INVERSE AND DETERMINANT FOR SUM OF CLASSES I AND J
      DO 10 JK=1,IVSZ

```

RHT00010
 RHT00020
 RHT00030
 RHT00040
 RHT00050
 RHT00060
 RHT00070
 RHT00080
 RHT00090
 RHT00100
 RHT00110
 RHT00120
 RHT00130
 RHT00140
 RHT00150
 RHT00160
 RHT00170
 RHT00180
 RHT00190
 RHT00200
 RHT00210
 RHT00220
 RHT00230
 RHT00240
 RHT00250
 RHT00260
 RHT00270
 RHT00280
 RHT00290
 RHT00300
 RHT00310
 RHT00320
 RHT00330
 RHT00340
 RHT00350
 RHT00360
 RHT00370
 RHT00380
 RHT00390
 RHT00400
 RHT00410
 RHT00420
 RHT00430
 RHT00440
 RHT00450
 RHT00460
 RHT00470
 RHT00480
 RHT00490
 RHT00500
 RHT00510
 RHT00520
 RHT00530
 RHT00540
 RHT00550
 RHT00560
 RHT00570
 RHT00580
 RHT00590
 RHT00600
 RHT00610
 RHT00620
 RHT00630
 RHT00640
 RHT00650
 RHT00660
 RHT00670
 RHT00680
 RHT00690
 RHT00700
 RHT00710
 RHT00720
 RHT00730
 RHT00740
 RHT00750
 RHT00760
 RHT00770
 RHT00780
 RHT00790

FILE: HMTCHR

```

      IF (IFULL.EQ.1) GO TO 8
      WRKRY(ICV2+IK-1)=COVMT2(IK,J)
      WRKRY(IW1+IK-1)=COVMT2(IK,J)+COVMT2(IK,I)
      GO TO 10
      A WRKRY(ICV2+IK-1)=COVMTX(IK,J)
      WRKRY(IW1+IK-1)=COVMTX(IK,J)+COVMTX(IK,I)
10  CONTINUE
      CALL COLINV(WRKRY(ICV2),NF,IERR,3,DET2)
      DET2=1./DET2
      IF (IERR.EQ.0) GO TO 15
      WRITE(4,100)J
      GO TO 50
15  CALL COLINV(WRKRY(IW1),NF,IERR,3,DET3)
      DET3=1./DET3
      IF (IERR.EQ.0) GO TO 16
      WRITE(4,200)I,J
      GO TO 50
16  IF (IFULL.NE.1) GO TO 1A
      DO 17 IK=1,NOFET2
17  T(IK)=AVENTX(IK,I)-AVENTX(IK,J)
      GO TO 25
1A  DO 20 IK=1,NOFET4
20  T(IK)=AVFMT2(IK,I)-AVFMT2(IK,J)
      MULTIPLY T TRANSPOSE TIMES WRKRY
      CONTINUE
      CALL MT4(WRKRY(IW1),T,WRKRY(IW2),NF,NF,1,1)
      RNUM=0.0
      DO 24 IK=1,NF
24  RNUM=RNUM+WRKRY(IW2+IK-1)*T(IK)
      DIVTAB(NM)=DEXP(-.25*RNUM-.5*DLLOG(DET3/(2.*NF
      *DSQRT(DET1*DET2))))
      SMSR=SMSR+WEIGHT(NM)*DIVTAB(NM)
      IF (IPART.LT.0) GO TO 50
C*
C*  COMPUTE PARTIALS
C*
      CALL MT4(WRKRY(IW2),BMAT,WRKRY(IW3),1,NOFET4,NOFET2,0)
      DO 30 IK=1,VAR522
30  WRKRY(IK+IW4-1)=COVMTX(IK,J)+COVMTX(IK,I)
      CALL MT4(WRKRY(IW4),WRKRY(IW3),WRKRY(IW2),NOFET2,NOFET2,1,1)
      DO 35 IK=1,NOFET2
35  WRKRY(IW3+IK-1)=AVENTX(IK,I)-AVENTX(IK,J)-WRKRY(IW2+IK-1)
      CALL MT4(T,WRKRY(IW3),WRKRY(IW5),NOFET4,1,NOFET2,0)
      CALL MT4(WRKRY(IW1),WRKRY(IW5),WRKRY(IW6),NOFET4,NOFET4,NOFET2,1)
      DO 40 IK=1,10
40  WRKRY(IW6+IK-1)=WRKRY(IW6+IK-1)/2
      M=T
      IC=ICV1
41  DO 45 IK=1,VAR524
      L=IK-1
45  WRKRY(IW4+L)=WRKRY(IW1+L)-WRKRY(IC+L)/2
      CALL MT4(WRKRY(IW4),BMAT,WRKRY(IW7),NOFET4,NOFET4,NOFET2,1)
      CALL MT4(WRKRY(IW7),COVMTX(1,M),WRKRY(IW5),NOFET4,NOFET2)
      DO 43 IK=1,10
      L=IK-1
43  WRKRY(IW6+L)=WRKRY(IW6+L)+WRKRY(IW5+L)
      IF (M.EQ.0) GO TO 46
      M=L
      IC=ICV2
      GO TO 41
46  DO 47 IK=1,10
      L=IK-1
      PARTLS(IK)=PARTLS(IK)-WEIGHT(NM)*DIVTAB(NM)*WRKRY(IW6+L)/NOCLS2
47  CONTINUE
50  CONTINUE
60  CONTINUE
      SMSR=SMSR/NOCLS2
      DETUN=J
100 FORMAT(' COVAR FOR CLASS',I3,' IS NOT POSITIVE DEFINITE')
200 FORMAT(' COVAR FOR SUM OF CLASSES',I24,' IS NOT POSITIVE DEF.')
300 FORMAT(' NOT ENOUGH WORK AREA AVAILABLE IN HMTCHR -- IWRK52=',I5)
      END

```

HMT00800
 HMT00810
 HMT00820
 HMT00830
 HMT00840
 HMT00850
 HMT00860
 HMT00870
 HMT00880
 HMT00890
 HMT00900
 HMT00910
 HMT00920
 HMT00930
 HMT00940
 HMT00950
 HMT00960
 HMT00970
 HMT00980
 HMT00990
 HMT01000
 HMT01010
 HMT01020
 HMT01030
 HMT01040
 HMT01050
 HMT01060
 HMT01070
 HMT01080
 HMT01090
 HMT01100
 HMT01110
 HMT01120
 HMT01130
 HMT01140
 HMT01150
 HMT01160
 HMT01170
 HMT01180
 HMT01190
 HMT01200
 HMT01210
 HMT01220
 HMT01230
 HMT01240
 HMT01250
 HMT01260
 HMT01270
 HMT01280
 HMT01290
 HMT01300
 HMT01310
 HMT01320
 HMT01330
 HMT01340
 HMT01350
 HMT01360
 HMT01370
 HMT01380
 HMT01390
 HMT01400
 HMT01410
 HMT01420
 HMT01430
 HMT01440
 HMT01450
 HMT01460
 HMT01470
 HMT01480
 HMT01490
 HMT01500
 HMT01510

FILE: BSTCHK

```

C      SUBROUTINE BSTCHK(NOREST)
C      IMPLICIT INTEGER (A-M,0-Z)
C      -----
C      CALL..    CALL BSTCHK(NOREST)
C      ARGS..    NOREST - NO OF FEATURES-TUPLES TO ANALYZE
C      REQUIRES.. COMMON /INFORM/
C      PURPOSE..  CHECKS VALIDITY OF REQUESTED FEATURE-TUPLES
C      RETURNS..  CORRECT FEATURE-TUPLE QUE
C      -----
C      CALL..    CALL SHWCHK(COMBUF,CPTR)
C      ARGS..    COMBUF - SHOW REQUEST QUE
C      PURPOSE..  CHECKS VALIDITY OF SHOW REQUESTS
C      RETURNS..  CORRECT SHOW REQUEST QUE
C      -----
C      DIMENSION COMBUF(1)
C      INCLUDE COMPK1,LIST
C      -----
C      DIMENSION INVERT(30)
C      INCLUDE COMPK7,LIST
C      COMMON/INFORM/NOCLS?,NOSUB?,NOFFT2,VARSZ2,TOTVT2,NOFLD2,
C      *      AVAR2,COVAR2,CLS1D2,SUBNO2,SURDS2,FLDSY2,VERTX2,
C      *      FFTVC2(30),SURVC2(75),SUBPTR(75),CLSV2(60),
C      *      KFPPTS(60),NOGRP,GWPNAM(60),GRPDEX(61),
C      *      GWPCHK(61),GRGUPS(124)
C      COMMON/FSL/CFAC,TOTMSR,SEPMSR,PRCKEY,CHKEY,INCFET,
C      *      INCVEC(30),ICOUNT,SETLGT,EVALHF(100),FFTVC4(30)
C      *      ,NOFFT4,VAWS74,COMBAS,DTAR4,WGMS14,BESTVC(10),DVSIZ
C      *      ,STATKY,ADRESN,ADRESF,ADHSH1,ADRSN2
C      *      ,INTEGR,ADHESN,ADHESF,ADHSH1,ADRSN2,STATKY
C      DOUBLE PRECISION CFAC,TOTMSR,SEPMSR
CSEMD
      II = 0
      DO 30 I=1,NOREST
      J = BESTVC(II)
      IF(J.GT.NOFFT2)GO TO 10
      II = II+1
      BESTVC(II) = J
      GO TO 30
10  WRITE (6,201)J
20  FORMAT('AA07A0  'HTEST'...',13,' IS GREATER THAN OR EQUAL TO NO. OF F
*FEATURES IN GIVEN DATA...[IGNORED]')
30  CONTINUE
      NOREST = II
C      RETURN
C      -----
C      ENTRY EVLCHK(COMBUF,CPTR)
C      DO 32 I=1,NOFFT2
32  INVERT(I)=0
      DO 33 I=1,NOFFT2
      K = FFTVC2(I)
33  INVERT(K)=1
      K = 0

```

HST00010
 HST00020
 HST00030
 HST00040
 HST00050
 HST00060
 HST00070
 HST00080
 HST00090
 HST00100
 HST00110
 HST00120
 HST00130
 HST00140
 HST00150
 HST00160
 HST00170
 HST00180
 HST00190
 HST00200
 HST00210
 HST00220
 HST00230
 HST00240
 HST00250
 HST00260
 HST00270
 HST00280
 HST00290
 HST00300
 HST00310
 HST00320
 HST00330
 HST00340
 HST00350
 HST00360
 HST00370
 HST00380
 HST00390
 HST00400
 HST00410
 HST00420
 HST00430
 HST00440
 COM00010
 COM00020
 COM00030
 COM00040
 COM00050
 COM00060
 HST00520
 HST00530
 HST00540
 HST00550
 HST00560
 HST00570
 HST00580
 HST00590
 HST00600
 HST00610
 HST00620
 HST00630
 HST00640
 HST00650
 HST00660
 HST00670
 HST00680
 HST00690
 HST00700
 HST00710
 HST00720
 HST00730
 HST00740
 HST00750
 HST00760
 HST00770
 HST00780
 HST00790
 HST00800

FILE: BSTCHK

I = 1	PST00810
40 IF (I.GE.CPTR) GO TO 100	RST00820
N = COMBUF(I)	RST00830
J = I+N	RST00840
I = I+1	RST00850
LAST = 0	RST00860
DO 50 L=I,J	RST00870
LL = COMBUF(L)	RST00880
IF (LL.LE.LAST.OR.INVERT(LL).EQ.0.OR.LL.GT.FETVC2(NOFET2)) GO TO 80	RST00890
10 LAST = LL	RST00900
K = K+1	RST00910
COMBUF(K) = N	RST00920
DO 60 L=I,J	RST00930
K = K+1	RST00940
60 COMBUF(K) = COMBUF(L)	RST00950
70 I = J+1	RST00960
GO TO 40	RST00970
C 80 WRITE (6,90) (COMBUF(LL),LL=I,J)	RST00980
90 FORMAT(' INVALID EVALUATE REQUEST.....',I4)	RST00990
GO TO 70	RST01000
C 100 CPTR = K	RST01010
C RETURN	RST01020
END	RST01030
	RST01040
	RST01050
	RST01060
	RST01070

FILE: COLINV

SUBROUTINE COLINV(S,N,IERR,IND,DET)
DOUBLE PRECISION S,SUR,DET

PURPOSE
INVERT A GIVEN SYMMETRIC POSITIVE DEFINITE MATRIX (S) BY
COMPUTING A TRIANGULAR FACTORIZATION (R), INVERTING R TO
OBTAIN A, AND THEN FINDING THE INVERSE OF S.
 $S=R*(TRANSPOSE\ OF\ R)$, $A=(INVERSE\ OF\ R)$,
 $(INVERSE\ OF\ S)=(TRANSPOSE\ OF\ A)*A$.

ARGUMENTS
S - LOWER TRIANGULAR PART OF THE GIVEN SYMMETRIC MATRIX
STORED ROWWISE IN $N*(N+1)/2$ SUCCESSIVE STORAGE LOCATIONS.
ON RETURN S CONTAINS THE LOWER TRIANGULAR MATRIX R,
THE LOWER TRIANGULAR MATRIX A, OR THE LOWER TRIANGULAR
PART OF THE INVERSE OF S, DEPENDING ON VALUE
ASSIGNED TO IND.
N - THE NUMBER OF ROWS OR COLUMNS IN GIVEN MATRIX.
IERR - RESULTING ERROR PARAMETER CODED AS FOLLOWS
IERR=0 - NO ERROR
IERR=-1 - MATRIX S IS NOT POSITIVE DEFINITE
IND - PARAMETER INDICATING WHICH MATRIX IS RETURNED --
IND=1 - MATRIX R
IND=2 - MATRIX A
IND=3 - MATRIX (INVERSE OF S)
DET - DETERMINANT OF RETURNED MATRIX.

REFERENCE
"INVERSION OF SYMMETRIC POSITIVE DEFINITE MATRICES," BY
J.K. BRYAN AND D.L. TERBE, E.E. DEPT., UNIV. OF MISSOURI, 1971.

DIMENSION S(1)
IERR = 0
LS = 0
N2 = 1
DET = 1.0
DO 70 I=1,N
N2 = N2 + I - 1
M = I*(I+1)/2
L = N2
DO 40 J=I,N
L = L + J - 1
IF(I.EQ.1) GO TO 20
M1 = J*(J-1)/2 + 1
L1 = L - 1
N1 = N2
DO 10 K=M1,L1
S(L) = S(L) - S(N1)*S(K)
10 N1 = N1 + 1
20 IF(J.EQ.1) GO TO 30
S(L) = S(L)/S(M)
GO TO 40
30 IF(S(L).LE.0.0) GO TO 100
S(L) = DSORT(S(L))
DET = DET*S(L)
40 CONTINUE
IF(IND.EQ.1) GO TO 70
S(M) = 1./S(M)
IF(I.EQ.1) GO TO 70
I1 = I - 1
DO 60 J=1,I1
K1 = I - J
L1 = LS
LS = LS + 1
M1 = J*(J-1)/2 + 1
J1 = J - 1
SUR = 0.0
DO 50 K=1,K1
L1 = L1 + 1
J1 = J1 + 1
M1 = M1 + J1 - 1
50 SUR = SUR + S(L1)*S(M1)
60 S(LS) = -S(M)*SUR
70 LS = LS + 1
IF(IND.EQ.2 .OR. IND.EQ.3) DET=1./DET
IF(IND.EQ.1 .OR. IND.EQ.2) RETURN
DO 90 I=1,N
M = I*(I-1)/2 + 1
L = M

COL00010
COL00020
COL00030
COL00040
COL00050
COL00060
COL00070
COL00080
COL00090
COL00100
COL00110
COL00120
COL00130
COL00140
COL00150
COL00160
COL00170
COL00180
COL00190
COL00200
COL00210
COL00220
COL00230
COL00240
COL00250
COL00260
COL00270
COL00280
COL00290
COL00300
COL00310
COL00320
COL00330
COL00340
COL00350
COL00360
COL00370
COL00380
COL00390
COL00400
COL00410
COL00420
COL00430
COL00440
COL00450
COL00460
COL00470
COL00480
COL00490
COL00500
COL00510
COL00520
COL00530
COL00540
COL00550
COL00560
COL00570
COL00580
COL00590
COL00600
COL00610
COL00620
COL00630
COL00640
COL00650
COL00660
COL00670
COL00680
COL00690
COL00700
COL00710
COL00720
COL00730
COL00740
COL00750
COL00760
COL00770
COL00780
COL00790

FILE: COLINV

```
DO 90 J=I,N
  L2 = L
  L = L + J - 1
  SUB = 0.0
  DO 80 K=J,N
    L2 = L2 + K - 1
    M1 = L2 + J - 1
    80 SUB = SUB + S(L2)*S(M1)
  90 S(L) = SUB
  DET = DET*DET
  RETURN
100 IERR = -1
  RETURN
END
```

COL00800
COL00810
COL00820
COL00830
COL00840
COL00850
COL00860
COL00870
COL00880
COL00890
COL00900
COL00910
COL00920
COL00930

FILE: CONVRT

C	SUBROUTINE CONVRT(NOSPAC,IFLG)	CON00010
C	THE PURPOSE OF THIS ROUTINE IS TO CONVERT EBCDIC CHARS TO COMPUTATIONAL	CON00020
C	OR TO CONVERT COMPUTATIONAL CHARS TO EBCDIC CHARS	CON00030
C	LOGICAL*1 LTEST(4),LSPAC(4)	CON00040
C	EQUIVALENCE (ITEST,LTEST(4)), (ICHAR,LSPAC(4))	CON00050
C	DATA BCD0/ZF0/,BCD9/ZF9/	CON00060
C	ITEST = 0	CON00070
C	ICHAR = NOSPAC	CON00080
C	LTEST(4) = LSPAC(4)	CON00090
C	CK FOR FLAG NOT 0, IMPLYING CHAR IS EBCDIC	CON00100
C	IF (IFLG .NE. 0) GO TO 1060	CON00110
C	CK FOR COMPUTATIONAL NUMBER & CONVERT TO EBCDIC	CON00120
C	IF (ITEST .LT. 0 .OR. ITEST .GT. 9) GO TO 9999	CON00130
C	ITEST = ITEST + BCD0	CON00140
C	LSPAC(4) = LTEST(4)	CON00150
C	NOSPAC = ICHAR	CON00160
C	RETURN	CON00170
C	CK FOR EBCDIC NUMBER TO CONVERT TO COMPUTATIONAL NUMBER	CON00180
C	1060 IF (ITEST .LT. BCD0 .OR. ITEST .GT. BCD9) GO TO 9999	CON00190
C	ITEST = ITEST - BCD0	CON00200
C	LSPAC(4) = LTEST(4)	CON00210
C	NOSPAC = ICHAR	CON00220
C	RETURN	CON00230
C	ERROR	CON00240
C	9999 WRITE (6,9999)	CON00250
C	9999 FORMAT (1X,'ERROR - NON-NUMERIC CHARACTER INPUT')	CON00260
C	RETURN	CON00270
C	END	CON00280
		CON00290
		CON00300
		CON00310
		CON00320
		CON00330
		CON00340
		CON00350

FILE: CURIC

SUBROUTINE CURIC (XX,YY,XMIN2)	CUB00010
DOUBLE PRECISION	CUB00020
1 XX(4) ,YY(4) ,ARRAY(4,4) ,COEFF(4) ,CONST(4)	CUB00030
2 TEMP ,ADD ,F ,XMIN2	CUB00040
DO 14 I = 1,4	CUB00050
ARRAY(I,1) = 1.000	CUB00060
ARRAY(I,2) = XX(I)	CUB00070
ARRAY(I,3) = XX(I)**2	CUB00080
ARRAY(I,4) = XX(I)**3	CUB00090
14 CONST(I) = YY(I)	CUB00100
15 DO 6 K2 = 1,3	CUB00110
KK = K2 + 1	CUB00120
L = K2	CUB00130
DO 7 I = KK,4	CUB00140
IF (DABS(ARRAY(I,K2)) - DABS(ARRAY(L,K2))) 7,7,20	CUB00150
20 CONTINUE	CUB00160
L = I	CUB00170
7 CONTINUE	CUB00180
IF (L - K2) 8,8,21	CUB00190
21 CONTINUE	CUB00200
DO 9 J = K2,4	CUB00210
TEMP = ARRAY(K2,J)	CUB00220
ARRAY(K2,J) = ARRAY(L,J)	CUB00230
9 ARRAY(L,J) = TEMP	CUB00240
TEMP = CONST(K2)	CUB00250
CONST(K2) = CONST(L)	CUB00260
CONST(L) = TEMP	CUB00270
8 DO 6 I = KK,4	CUB00280
F = ARRAY(I,K2)/ARRAY(K2,K2)	CUB00290
ARRAY(I,K2) = 0.000	CUB00300
DO 10 J = KK,4	CUB00310
10 ARRAY(I,J) = ARRAY(I,J) - F*ARRAY(K2,J)	CUB00320
6 CONST(I) = CONST(I) - F*CONST(K2)	CUB00330
COEFF(4) = CONST(4)/ARRAY(4,4)	CUB00340
I = 3	CUB00350
11 II = I + 1	CUB00360
ADD = 0.000	CUB00370
DO 12 J = II,4	CUB00380
12 ADD = ADD + ARRAY(I,J)*COEFF(J)	CUB00390
COEFF(I) = (CONST(I) - ADD)/ARRAY(I,I)	CUB00400
I = I - 1	CUB00410
IF (I) 22,22,11	CUB00420
22 CONTINUE	CUB00430
F = COEFF(3)**2 - 3.000*COEFF(2)*COEFF(4)	CUB00440
IF (F) 16,23,23	CUB00450
23 CONTINUE	CUB00460
IF (DABS(COEFF(4)) - 1.00-15) 17,17,24	CUB00470
24 CONTINUE	CUB00480
XMIN2 = (-COEFF(3) + DSQRT(F))/(3.000*COEFF(4))	CUB00490
RETURN	CUB00500
17 IF (DABS(COEFF(3)) - 1.00-15) 16,16,25	CUB00510
25 CONTINUE	CUB00520
XMIN2 = -COEFF(2)/(COEFF(3) + DSQRT(F))	CUB00530
RETURN	CUB00540
16 XMIN2 = 0.000	CUB00550
RETURN	CUB00560
END	CUB00570

FILE: DAVDN1

```

C*      SUBROUTINE DAVDN1(FX,P,H)
C*
C*      DAVDN1 INITIALIZES THE H AND P ARRAYS USED IN DAVIDON.PROCEDURE
C*
C*      H***** MATRIX IS AN N*N MATRIX APPROXIMATING THE INVERSE
C*      MATRIX OF PARTIAL DERIVATIVES. IT IS INITIALIZED
C*      HERE AS THE IDENTITY MATRIX AND STORED ON SCRATCH
C*      FILE. THE H MATRIX IS UPDATED AFTER EACH CYCLE THRU
C*      THE SEARCH PROCEDURE OF SUBROUTINE DAVDN2.
C*
C*      P***** VECTOR CORRESPONDING TO SEARCH DIRECTION
C*      (H-MATRIX TIMES THE GRADIENT)
C*      DFDK** MINUS THE NORM OF THE GRADIENT WITH RESPECT TO THE
C*      H-MATRIX SQUARED.
C*      FX**** GRADIENT OF FUNCTION TO BE MINIMIZED
C*
C*      * H * AND * FX* SAVED ON SCRATCH FILE TO BE USED IN DAVDN3
C*
C*      INCLUDE COM47.LIST
C*      COMMON/FSL/CFAC,TOTMSR,SEPMSR,PRCKEY,CRIKEY,INCFET,
C*      INCVEC(30),ICOUNT,SETWGT,EVALBF(100),FETVC4(30)
C*      ,INFET4,VARSZ4,CORBAS,DTAB4,WGHS14,RESTVC(10),DIVSIZ
C*      ,STATKY,ADRES0,ADRESP,ADRESF,ADRSH1,ADRSH2
C*      ,INTEGER ADRES0,ADRESP,ADRESF,ADRSH1,ADRSH2,STATKY
C*      DOUBLE PRECISION CFAC,TOTMSR,SEPMSR
C*
C*      DOUBLE PRECISION FX(1),P(1),H(1),DFDK
C*      DOUBLE PRECISION CAYMIN,FII,CCAY
C*      INTEGER AD
C*      COMMON/DVNBK/DFDK,CAYMIN,FII,CCAY,I10,I1DMEN,ITT,ICNT,N
C*      N2=N*2
C*      DFDK=0.0
C*      AD=ADRSH1
C*      DO 30 J=1,N
C*      P(1)=-FX(1)
C*      DFDK=DFDK+P(1)*FX(1)
C*      DO 20 J=1,N
C*      H(J)=0.0
C*      IF(I.EQ.J)H(J)=1.0
C*      CALL RWRITE(AD,H,N2,ISTAT)
C*      AD=AD+N2
C*      IF(ISTAT.EQ.1)GO TO 21
C*      IF(ISTAT.EQ.0)GO TO 30
C*      WRITE(6,100)ISTAT
C*      30 CONTINUE
C*
C*      SAVE FX ON SCRATCH FILE
C*
C*      CALL RWRITE(ADRESF,FX,N2,ISTAT)
C*      IF(ISTAT.EQ.1)GO TO 40
C*      IF(ISTAT.EQ.0)RETURN
C*      WRITE(6,100)ISTAT
C*      CALL CMERR
C*      100 FORMAT(/' ERROR ON DRUM FILE - SUBROUTINE DAVDN1---ISTAT=',I3)
C*      END

```

DAV00010
 DAV00020
 DAV00030
 DAV00040
 DAV00050
 DAV00060
 DAV00070
 DAV00080
 DAV00090
 DAV00100
 DAV00110
 DAV00120
 DAV00130
 DAV00140
 DAV00150
 DAV00160
 DAV00170
 DAV00180
 DAV00190
 COM00010
 COM00020
 COM00030
 COM00040
 COM00050
 COM00060
 DAV00210
 DAV00220
 DAV00230
 DAV00240
 DAV00250
 DAV00260
 DAV00270
 DAV00280
 DAV00290
 DAV00300
 DAV00310
 DAV00320
 DAV00330
 DAV00340
 DAV00350
 DAV00360
 DAV00370
 DAV00380
 DAV00390
 DAV00400
 DAV00410
 DAV00420
 DAV00430
 DAV00440
 DAV00450
 DAV00460
 DAV00470
 DAV00480
 DAV00490
 DAV00500

FILE: DAVDN2

```

SURROUTINE DAVDN2(XBAR,XBARS,P,WRK,IWKSZ,*)
DOUBLE PRECISION XPAR(N),XBARS(N),P(N),WRK(IWKSZ)
DOUBLE PRECISION FHAT(4),CAY(4),X(10),Y(10),
* CAYMIN,CCAY,DFDK,EPSCF,EPSCG,EX,F,FII,
* G1,G2,TEMP,XMIN1,XMIN2,XNEW
COMMON/DVNHLC/DFDK,CAYMIN,FII,CCAY,IID,IIDMEN,ITT,ICNT,N
DFDK***** MINUS THE NORM OF THE GRADIENT WITH RESPECT TO
THE H-MATRIX SQUARED
FII***** INITIAL VALUE OF FUNCTION TO BE MINIMIZED
IPART***** SET TO -1 SO PARTIALS WILL NOT BE COMPUTED
DURING SEARCH
N***** NUMBER OF VARIABLES
ITT***** COUNTER DENOTING THE NUMBER OF EVALUATIONS
OF THE FUNCTION TO BE MINIMIZED
P***** VECTOR CORRESPONDING TO SEARCH DIRECTION
(X-MATRIX TIMES THE GRADIENT)
XRARS***** NOMINAL VECTOR OF CONTROL PARAMETERS AT
THE START OF EACH CYCLE
XBARS***** PERTURBED VECTOR OF CONTROL PARAMETERS IN
ACCORDANCE WITH THE DAVIDON ALGORITHM AND
SEARCH PROCEDURES
G1 = 1.5D0 - DSQRT(1.25D0)
G2 = 1.0D0 - G1

THIS STARTS ONE DIMENSIONAL SEARCH ON K
INITIALIZE 1D SEARCH VARIABLES
EPSCF IS RELATIVE EPSILON FOR CUBIC FIT.
EPSCG IS RELATIVE EPSILON FOR GOLDEN SECTION.
KUBIC = 1 CUBIC FIT METHOD
KUBIC = 0 GOLDEN SECTION METHOD
KUBIC = -1 GOLDEN SECTION AND CUBIC FIT METHODS COMBINED

9038 IPART=-1
Y(10) = FII
JSW = -1
KUBIC=1
EPSCF=1.0D-3
CAY(1)=0.0D+0
FHAT(1) = FII
CAY(2)=CCAY
DO 9039 I=1,N
9039 XBARS(I)=XBAR(I)+CAY(2)*P(I)
CALL FIN2(FHAT(2),IPART,XBARS,WRK,IWKSZ,69999)
Y(2) = FHAT(2)
X(2)=CAY(2)
ITT=ITT+1
Y(1)=2.0D+0*(FHAT(2)-FHAT(1)-DFDK*CAY(2))/(CAY(2)*CAY(2))
IF(Y(1) - 1.0D-15)9040,1,1
1 CONTINUE
XMIN1=-DFDK/Y(1)
CAY(3)=XMIN1
DO 9041 I=1,N
9041 XBARS(I)=XBAR(I)+CAY(3)*P(I)
CALL FIN2(FHAT(3),IPART,XBARS,WRK,IWKSZ,69999)
ITT=ITT+1
GO TO 9043
9040 CAY(3)=CCAY*1.2D+0
XMIN1=CAY(3)
GO TO 9042
9043 Y(1)=FHAT(2)-FHAT(1)-DFDK*CAY(2)
Y(2)=FHAT(3)-FHAT(1)-DFDK*CAY(3)
X(1)=CAY(2)*CAY(2)/2.0D+0
X(2)=X(1)*CAY(2)/3.0D+0
X(3)=CAY(3)*CAY(3)/2.0D+0
X(4)=X(3)*CAY(3)/3.0D+0
Y(3)=X(3)*X(2)-X(1)*X(4)
Y(4)=Y(3)
IF(Y(3) - 1.0D+0)Y(4)=-Y(3)
XNEW = Y(4)
Y(4)=(X(2)*Y(2)-X(4)*Y(1))/Y(3)
Y(5)=(X(3)*Y(1)-X(1)*Y(2))/Y(3)
Y(6)=Y(4)*Y(4)-2.0D+0*DFDK*Y(5)
IF(Y(6))9044,2,2
2 CONTINUE
XMIN2 = -2.0D0*DFDK/(Y(4) + DSQRT(Y(6)))
IF(XMIN2 - 1.0D-15)9044,3,3
3 CONTINUE
IF(DABS(XMIN2-XMIN1) - EPSCF*XMIN1)4,4,9045

```

DAV00010
 DAV00020
 DAV00030
 DAV00040
 DAV00050
 DAV00060
 DAV00070
 DAV00080
 DAV00090
 DAV00100
 DAV00110
 DAV00120
 DAV00130
 DAV00140
 DAV00150
 DAV00160
 DAV00170
 DAV00180
 DAV00190
 DAV00200
 DAV00210
 DAV00220
 DAV00230
 DAV00240
 DAV00250
 DAV00260
 DAV00270
 DAV00280
 DAV00290
 DAV00300
 DAV00310
 DAV00320
 DAV00330
 DAV00340
 DAV00350
 DAV00360
 DAV00370
 DAV00380
 DAV00390
 DAV00400
 DAV00410
 DAV00420
 DAV00430
 DAV00440
 DAV00450
 DAV00460
 DAV00470
 DAV00480
 DAV00490
 DAV00500
 DAV00510
 DAV00520
 DAV00530
 DAV00540
 DAV00550
 DAV00560
 DAV00570
 DAV00580
 DAV00590
 DAV00600
 DAV00610
 DAV00620
 DAV00630
 DAV00640
 DAV00650
 DAV00660
 DAV00670
 DAV00680
 DAV00690
 DAV00700
 DAV00710
 DAV00720
 DAV00730
 DAV00740
 DAV00750
 DAV00760
 DAV00770
 DAV00780
 DAV00790

FILE: DAVDN2

```

4 CONTINUE
  CAYMIN=XMIN2
  GO TO 9045
9045 CAY(4)=XMIN2
  GO TO 9048
9044 CAY(4)=CAY(3)*1.5D+0
9048 DO 9047 I=1,N
9047 XBARS(I)=XBAR(I)+CAY(4)*P(I)
  CALL FINT2(FHAT(4),IPART,XBARS,WRK,IWKSZ,&9999)
  ITT=ITT+1
  DO 9050 J=1,3
  DO 9050 I=1,3
  IF(CAY(I) - CAY(I+1))9050,5,5
5 CONTINUE
  TEMP = CAY(I)
  CAY(I) = CAY(I+1)
  CAY(I+1) = TEMP
  TEMP = FHAT(I)
  FHAT(I) = FHAT(I+1)
  FHAT(I+1) = TEMP
9050 CONTINUE
  GO TO 542
9066 FII = Y(10)
  CAY(4)=X(9)
  FHAT(4)=Y(9)
  IJDMEN = -1
  R01 IPART=-1
  KURJC = 1
  IF(IID)888,888,6
6 CONTINUE
  KURIC = 0
  888 XMIN1 = 0.000
  CAY(1)=0.0+0
  FHAT(1) = FII
  IF(DEFK.GT.-1.0-12)GO TO 340
  IF(DEFK.GT.-1.0-9) GO TO 330
  IF(DEFK.GT.-1.0-5) GO TO 320
  IF(DEFK.GT.-1.0+1)GO TO 350
  EPSGS=1.0-2
  FPSCF=1.0-2
  GO TO 410
350 FPSCF=1.0-3
  EPSGS=1.0-3
  GO TO 410
320 FPSCF=1.0-5
  EPSGS=1.0-5
  GO TO 410
330 EPSCF = 1.0-2
  FPSCF = 1.0-2
  GO TO 410
340 EPSCF = 1.0-1
  FPSCF = 1.0-1
C
C C
C PART 1
  ESTABLISH GOLDEN SECTION IN WHICH FUNCTION IS UNIMODAL
410 IF(IID)7,7,9067
7 CONTINUE
  CAY(4)=CCAY
  DO 430 I=1,N
430 XBARS(I)=XBAR(I)+CAY(4)*P(I)
  CALL FINT2(FHAT(4),IPART,XBARS,WRK,IWKSZ,&9999)
  ITT = ITT + 1
9067 CONTINUE
  IF(FHAT(4) - FHAT(1))8,490,490
8 CONTINUE
  440 CAY(2)=CAY(4)
  FHAT(2)=FHAT(4)
  CAY(4) = CAY(2)/G1
  DO 450 I=1,N
450 XBARS(I)=XBAR(I)+CAY(4)*P(I)
  CALL FINT2(FHAT(4),IPART,XBARS,WRK,IWKSZ,&9999)
  ITT = ITT + 1
  IF(FHAT(4) - FHAT(2))440,9,9
9 CONTINUE
  460 CAY(3) = G2*(CAY(4) - CAY(1)) + CAY(1)
  DO 470 I=1,N
470 XBARS(I)=XBAR(I)+CAY(3)*P(I)
  CALL FINT2(FHAT(3),IPART,XBARS,WRK,IWKSZ,&9999)

```

DAV00800
 DAV00810
 DAV00820
 DAV00830
 DAV00840
 DAV00850
 DAV00860
 DAV00870
 DAV00880
 DAV00890
 DAV00900
 DAV00910
 DAV00920
 DAV00930
 DAV00940
 DAV00950
 DAV00960
 DAV00970
 DAV00980
 DAV00990
 DAV01000
 DAV01010
 DAV01020
 DAV01030
 DAV01040
 DAV01050
 DAV01060
 DAV01070
 DAV01080
 DAV01090
 DAV01100
 DAV01110
 DAV01120
 DAV01130
 DAV01140
 DAV01150
 DAV01160
 DAV01170
 DAV01180
 DAV01190
 DAV01200
 DAV01210
 DAV01220
 DAV01230
 DAV01240
 DAV01250
 DAV01260
 DAV01270
 DAV01280
 DAV01290
 DAV01300
 DAV01310
 DAV01320
 DAV01330
 DAV01340
 DAV01350
 DAV01360
 DAV01370
 DAV01380
 DAV01390
 DAV01400
 DAV01410
 DAV01420
 DAV01430
 DAV01440
 DAV01450
 DAV01460
 DAV01470
 DAV01480
 DAV01490
 DAV01500
 DAV01510
 DAV01520
 DAV01530
 DAV01540
 DAV01550
 DAV01560
 DAV01570
 DAV01580

FILE: DAVON2

	ITT = ITT + 1	DAV01590
	IF (FHAT(4) - FHAT(3))10,10,510	DAV01600
10	CONTINUE	DAV01610
475	FHAT(4)=FHAT(3)	DAV01620
	CAY(4)=CAY(3)	DAV01630
	CAY(3)=CAY(2)	DAV01640
	FHAT(3)=FHAT(2)	DAV01650
	CAY(2)=CAY(4) - CAY(3) + CAY(1)	DAV01660
	DO 480 I=1,N	DAV01670
480	XBARS(I)=XBAR(I)+CAY(2)*P(I)	DAV01680
	CALL FINT2(FHAT(2),IPART,XBARS,WKK,IWKSZ,&9999)	DAV01690
	ITT = ITT + 1	DAV01700
	IF (FHAT(2) - FHAT(1))510,11,11	DAV01710
11	CONTINUE	DAV01720
485	FHAT(4) = FHAT(2)	DAV01730
	CAY(4) = CAY(2)	DAV01740
490	IF (IDAVN.EQ.2) GO TO 8007	DAV01750
	CAY(2) = 61*(CAY(4) - CAY(1)) + CAY(1)	DAV01760
	IF (CAY(2) - 1.00-15)12,12,495	DAV01770
12	CONTINUE	DAV01780
8007	CAYMIN=0.0+0	DAV01790
	WRITE(6,492)	DAV01800
492	FORMAT(1X,46HMINIMUM IS AT ORIGIN - PROGRAM CANNOT CONTINUE)	DAV01810
	RETURN 1	DAV01820
495	DO 500 I=1,N	DAV01830
500	XBARS(I)=XBAR(I)+CAY(2)*P(I)	DAV01840
	CALL FINT2(FHAT(2),IPART,XBARS,WKK,IWKSZ,&9999)	DAV01850
	ITT = ITT + 1	DAV01860
	IF (FHAT(2) .LT. FHAT(1))GO TO 460	DAV01870
	GO TO 485	DAV01880
C	PART 2	DAV01890
C	SHRINK THE GOLDEN SECTION CONTAINING THE MINIMUM	DAV01900
510	IF (110MFN)13,13,9066	DAV01910
13	CONTINUE	DAV01920
514	IF (FHAT(3) - FHAT(2))14,525,520	DAV01930
14	CONTINUE	DAV01940
	CAY(1)=CAY(2)	DAV01950
	FHAT(1)=FHAT(2)	DAV01960
	CAY(2)=CAY(3)	DAV01970
	FHAT(2)=FHAT(3)	DAV01980
	CAY(3)=CAY(1)+CAY(4)-CAY(2)	DAV01990
	FX = CAY(3)	DAV02000
	JSW = 1	DAV02010
	GO TO 612	DAV02020
520	CAY(4) = CAY(3)	DAV02030
	CAY(4)=CAY(3)	DAV02040
	FHAT(4)=FHAT(3)	DAV02050
	CAY(3)=CAY(2)	DAV02060
	FHAT(3)=FHAT(2)	DAV02070
	CAY(2)=CAY(1)+CAY(4)-CAY(3)	DAV02080
	FX = CAY(2)	DAV02090
	JSW = -1	DAV02100
	GO TO 612	DAV02110
525	FHAT(1) = FHAT(2)	DAV02120
	FHAT(1)=FHAT(2)	DAV02130
	CAY(4)=CAY(3)	DAV02140
	FHAT(4)=FHAT(3)	DAV02150
	CAY(3) = 3.00+0*CAY(2) - CAY(3) - CAY(1)	DAV02160
	CAY(1)=CAY(2)	DAV02170
	CAY(2) = CAY(4) + CAY(1) - CAY(3)	DAV02180
	FX = CAY(2)	DAV02190
	JSW = 0	DAV02200
	GO TO 612	DAV02210
526	FHAT(3)=F	DAV02220
C	CHECK ON THE UNIMODALITY OF THE FUNCTION IN THE NEW INTERVAL	DAV02230
530	IF (FHAT(2) - FHAT(1))15,15,485	DAV02240
15	CONTINUE	DAV02250
	IF (FHAT(3) - FHAT(4))16,16,475	DAV02260
16	CONTINUE	DAV02270
C	PART 3	DAV02280
C	FIND MINIMUM BY EITHER GOLDEN SECTION OR CUBIC FIT TECHNIQUE	DAV02290
540	IF (CAY(3) - CAY(2)*(1.000+EPSGS))585,17,17	DAV02300
17	CONTINUE	DAV02310
		DAV02320
		DAV02330
		DAV02340
		DAV02350
		DAV02360
		DAV02370

FILE: DAVON2

```

      IF (KURIC.EQ.0) GO TO 514
      IF (CAY(4) - 1.0D-5) 510,542,542
542 CALL CURIC(CAY,FHAT,XMIN2)
      IF (XMIN2) 510,510,18
      18 CONTINUE
      IF (KURIC.GT.0) GO TO 2000
      IF (XMIN1) 580,580,19
      19 CONTINUE
      IF (XMIN2*(1.0D0-EPSCF) - XMIN1) 20,20,580
      20 CONTINUE
545 CAYMIN = (XMIN2 + XMIN1)/2.0D+0
      GO TO 9999
590 XMIN1 = XMIN2
      GO TO 510
595 CAYMIN = (CAY(3)+CAY(2))/2.0D+0
      GO TO 9999
C
612 DO 613 I=1,N
613 XHARS(I) = XHAR(I) + EX*P(I)
      CALL FINT2(F,IPART,XHARS,WRK,IWKSZ,&9999)
      ITT = ITT + 1
      IF (JSW) 21,21,536
      21 CONTINUE
      FHAT(2) = F
      IF (JSW) 540,22,22
      22 CONTINUE
      FX = CAY(3)
      JSW = 1
      GO TO 512
C
      REARRANGE Y(I) SO THEY ARE IN ASCENDING ORDER
C
2000 DO 2020 I = 1,4
      X(I) = CAY(I)
2020 Y(I) = FHAT(I)
      DO 2040 J = 1,3
      DO 2040 I = 1,3
      IF (Y(I) - Y(I+1)) 2040,23,23
      23 CONTINUE
      TEMP = Y(I)
      Y(I) = Y(I+1)
      Y(I+1) = TEMP
      TEMP = X(I)
      X(I) = X(I+1)
      X(I+1) = TEMP
2040 CONTINUE
3000 DO 3001 I=1,N
3001 XHARS(I) = XHAR(I) + XMIN2 * P(I)
      CALL FINT2(F,IPART,XHARS,WRK,IWKSZ,&9999)
      ITT = ITT + 1
3005 DO 3010 K = 1,4
      J = K
      IF (F.LT.Y(K)) GO TO 3020
3010 CONTINUE
      GO TO 4000
3020 I = 4
3025 IF (I.EQ.0) GO TO 3030
      Y(I) = Y(I-1)
      X(I) = X(I-1)
      I = I-1
      GO TO 3025
3030 Y(I) = F
      X(I) = XMIN2
3045 CALL CURIC(X,Y,XMIN2)
3050 IF (XMIN2) 24,4000,24
      24 CONTINUE
      IF (DARS(XMIN2-X(I)) - EPSCF*XMIN2) 25,25,3000
      25 CONTINUE
      CAYMIN = XMIN2
      GO TO 9999
4000 IF (IIDMEN) 26,26,9066
      26 CONTINUE
      KURIC = -1
      XMIN1 = 0.0D+0
      XMIN2 = 0.0D+0
      GO TO 514
9999 CAY=CAYMIN
      IIDMEN=110
C

```

DAV02380
 DAV02390
 DAV02400
 DAV02410
 DAV02420
 DAV02430
 DAV02440
 DAV02450
 DAV02460
 DAV02470
 DAV02480
 DAV02490
 DAV02500
 DAV02510
 DAV02520
 DAV02530
 DAV02540
 DAV02550
 DAV02560
 DAV02570
 DAV02580
 DAV02590
 DAV02600
 DAV02610
 DAV02620
 DAV02630
 DAV02640
 DAV02650
 DAV02660
 DAV02670
 DAV02680
 DAV02690
 DAV02700
 DAV02710
 DAV02720
 DAV02730
 DAV02740
 DAV02750
 DAV02760
 DAV02770
 DAV02780
 DAV02790
 DAV02800
 DAV02810
 DAV02820
 DAV02830
 DAV02840
 DAV02850
 DAV02860
 DAV02870
 DAV02880
 DAV02890
 DAV02900
 DAV02910
 DAV02920
 DAV02930
 DAV02940
 DAV02950
 DAV02960
 DAV02970
 DAV02980
 DAV02990
 DAV03000
 DAV03010
 DAV03020
 DAV03030
 DAV03040
 DAV03050
 DAV03060
 DAV03070
 DAV03080
 DAV03090
 DAV03100
 DAV03110
 DAV03120
 DAV03130
 DAV03140
 DAV03150
 DAV03160

FILE: DAVDN2

```
C THIS ENDS ONE DIMENSIONAL SEARCH ON K
C
30 DO 31 I=1,N
   P(I)=CAYMIN*P(I)
   XRAR(I)=XBAR(I)*P(I)
31 CONTINUE
   RETURN
   END
```

DAV03170
DAV03180
DAV03190
DAV03200
DAV03210
DAV03220
DAV03230
DAV03240

FILE: DAVDN3

```

C* SURROUTINE DAVDN3(FX,FX1,P,H,HY)
C* DAVDN3 - UPDATES THE H MATRIX AND THE P MATRIX FOR THE NEXT
C* CYCLE THROUGH THE SEARCH.
C* - THE H MATRIX IS BROUGHT IN TO CORE ONE ROW AT A TIME.
C* SCRATCH FILES 7 AND 28 ARE USED TO STORE AND UPDATE
C* THIS MATRIX.
C* N***** - NUMBER OF VARIABLES
C* FX***** - GRADIENT OF FUNCTION BEING MINIMIZED
C* (PARTIAL DER. WITH RESPECT TO NEW H-MATRIX)
C* FX1*** - VALUE OF GRADIENT AT THE BEGINNING OF THE CYCLE.
C* (PARTIAL DER. WITH RESPECT TO OLD H-MATRIX)
C* H***** - THE H MATRIX APPROXIMATES THE INVERSE MATRIX OF
C* PARTIAL DERIVATIVES.
C* HY***** - VECTOR USED IN UPDATING THE H-MATRIX
C* P***** - VECTOR CORRESPONDING TO SEARCH DIRECTION
C* (H-MATRIX TIMES THE GRADIENT)
C*
C* INCLUDE COMPK7.LIST
C* COMMON/FSL/CFAC,TOTMSR,SEPMSR,PRCKEY,CRICKY,INCFET,
C* INCVFC(30),ICOUNT,SFTWGT,EVALPF(100),FETVC4(30)
C* NOFET4,VAR574,CORHAS,OTAR4,WGHS14,RESTVC(10),DIVSIZ
C* STATKY,ADRES0,ADRES1,ADRES2,ADRESH1,ADRESH2
C* INTEGER ADRES0,ADRES1,ADRES2,ADRESH1,ADRESH2,STATKY
C* DOUBLE PRECISION CFAC,TOTMSR,SEPMSR
C*
C* SEND
C* DOUBLE PRECISION DFDK,CAYMIN,CCAY,FII
C* DOUBLE PRECISION SIGYI,YHY,DELXFX
C* DOUBLE PRECISION FX(N),FX1(N),P(N),H(N),HY(N)
C* COMMON/DVNHK/DFDK,CAYMIN,FII,CCAY,IID,IIDMEN,ITT,ICNT,N
C* N2=N*2
C* KAD=ADRESH1
C* LAD=ADRESH2
C* IF(MOD(ICNT,2).EQ.1)GO TO 10
C* KAD=ADRESH2
C* LAD=ADRESH1
C* 10 CONTINUE
C* CALL RREAD(ADRESF,FX1,N2,ISTAT)
C* UNIVAC CHECK DRUM STATUS
C* DELXFX=0.0
C* JAD=KAD
C* DO 40 I=1,N
C* DELXFX=DELXFX + P(I)*FX(I)/CAYMIN
C* HY(I)=0.0
C* READ ONE ROW OF THE H MATRIX
C* CALL RREAD(LAD,H,N2,ISTAT)
C* UNIVAC CHECK DRUM STATUS
C* JAD=JAD+N2
C* DO 30 J=1,N
C* 30 HY(I)=HY(I) + H(J)*(FX(J)-FX1(J))
C* 40 CONTINUE
C* YHY=0.0
C* SIGYI=0.0
C* DO 50 I=1,N
C* SIGYI=SIGYI + P(I)*(FX(I)-FX1(I))
C* 50 YHY=YHY + (FX(I)-FX1(I))*HY(I)
C* IF(SIGYI.LT.1.0-36 .OR. YHY .LT. 1.0-36)GO TO 60
C* DO 65 I=1,N
C* FX1(I)=0.0
C* CALL RREAD(KAD,H,N2,ISTAT)
C* UNIVAC CHECK DRUM STATUS
C* KAD=KAD+N2
C* UPDATE THIS ROW OF THE H MATRIX
C* 55 DO 60 J=1,N
C* H(J) = H(J) + P(I)*P(J)/SIGYI - HY(I)*HY(J)/YHY
C* 60 CONTINUE
C* DO 61 J=1,N
C* 61 CONTINUE
C* WRITE UPDATED H
C* CALL RWRITE(LAD,H,N2,ISTAT)
C* LAD=LAD+N2
C* UNIVAC CHECK DRUM STATUS
C* 65 CONTINUE

```

DAV00010
 DAV00020
 DAV00030
 DAV00040
 DAV00050
 DAV00060
 DAV00070
 DAV00080
 DAV00090
 DAV00100
 DAV00110
 DAV00120
 DAV00130
 DAV00140
 DAV00150
 DAV00160
 DAV00170
 DAV00180
 DAV00190
 DAV00200
 DAV00210
 DAV00220
 DAV00230
 DAV00240
 DAV00250
 DAV00260
 DAV00270
 DAV00280
 DAV00290
 DAV00300
 DAV00310
 DAV00320
 DAV00330
 DAV00340
 DAV00350
 DAV00360
 DAV00370
 DAV00380
 DAV00390
 DAV00400
 DAV00410
 DAV00420
 DAV00430
 DAV00440
 DAV00450
 DAV00460
 DAV00470
 DAV00480
 DAV00490
 DAV00500
 DAV00510
 DAV00520
 DAV00530
 DAV00540
 DAV00550
 DAV00560
 DAV00570
 DAV00580
 DAV00590
 DAV00600
 DAV00610
 DAV00620
 DAV00630
 DAV00640
 DAV00650
 DAV00660
 DAV00670
 DAV00680
 DAV00690
 DAV00700
 DAV00710
 DAV00720
 DAV00730
 DAV00740
 DAV00750
 DAV00760
 DAV00770
 DAV00780
 DAV00790

FILE: DAVDN3

	DFDK=0.0	DAV00800
	DO 70 I=1,N	DAV00810
	P(I)=FX(I)	DAV00820
	70 DFDK=DFDK+FX(I)*P(I)	DAV00830
C*		DAV00840
C*	SAVE LATEST PARTIALS ON SCRATCH FILE	DAV00850
C*		DAV00860
C	CALL RWRITE(ADRESF,FX,N2,ISTAT)	DAV00870
C	UNIVAC CHECK DRUM STATUS	DAV00880
	RETURN	DAV00890
AN	WRITE(A,300)SIGYI,YHY	DAV00900
	RETURN	DAV00910
C*		DAV00920
300	FORMAT(' *DAVDN3*--EITHER SIGYI OR YHY TOO CLOSE TO ZERO TO UPDATE	DAV00930
	* M MATRIX--SIGYI=*,E15.7,* YHY=*,E15.7)	DAV00940
C	UNIVAC CHECK DRUM STATUS	DAV00950
	END	DAV00960

FILE: DAVIDN

```

      SURROUTINE DAVIDN(COVMTX,AVEMTX,DIVTAB,WEIGHT,COVMT2,AVEMT2,S,S2, DAV00010
      HMAT,PARTLS,WKRY,IWKSZ) DAV00020
      ..... DAV00030
C* DAV00040
C* DAVIDN IS THE DRIVER ROUTINE FOR THE DAVIDON-FEETCHER-POWELL PRO DAV00050
C* DAV00060
C* INCLUDE COMMK1.LIST DAV00070
C* INCLUDE COM-K7.LIST DAV00080
C* COMMON/INFORM/NOCLS2,NOSUR2,NOFET2,VARSZ2,TOTVT2,NOFLD2, DAV00090
C*   AVA22,COVAR2,CLSID2,SUHN02,SURDS2,FLUSV2,VENTX2, DAV00100
C*   FETVC2(30),SUMVC2(75),SURPTH(75),CLSVC2(60), DAV00110
C*   KFPPTS(60),NOGRP,GMPNAM(60),GRPDEX(61), DAV00120
C*   GRPCHK(61),GROUPS(124) DAV00130
C* COMMON/FSL/CFAC,TOTMSH,SEPM5H,PCHKEY,CRIKEY,INCFET, DAV00140
C*   INVEC(30),ICOUNT,SETWGT,EVALHF(100),FETVC4(30) DAV00150
C*   NOFET4,VARSZ4,CORHAS,UTAH4,WGHS14,MESTVC(10),DIVSI2 DAV00160
C*   ,STATKY,ADHESD,ADHESP,ADHESF,ADHSH1,ADHSH2 DAV00170
C*   INTEGER ADHESD,ADHESP,ADHESF,ADHSH1,ADHSH2,STATKY DAV00180
C*   DOUBLE PRECISION CFAC,TOTMSH,SEPM5H DAV00190
CSEND DAV00200
C* EQUIVALENCE (IRLOCK(2),PCHKEY) DAV00210
C* DOUBLE PRECISION HMAT(NOFET4,NOFET2) DAV00220
C* DIMENSION WKRY(1) DAV00230
C* DIMENSION COVMTX(1), AVEMTX(1), WEIGHT(1), S(1) DAV00240
C* INTEGER FETVC2,FETVC4,BMKEY,CRIKEY,PCHKEY DAV00250
C* INTEGER SETWGT DAV00260
C* COMMON/DVNRK/DFDK,CAYMIN,FII,CCAY,I10,I1DMEN,ITT,ICNT,N DAV00270
C* DOUBLE PRECISION DELF,FII,CAYMIN,DFDK,CCAY,PARTLS(1),SMSR DAV00280
C* DOUBLE PRECISION DIVTAB(1),COVMT2(1),AVEMT2(1),S2(1) DAV00290
C* CCAY=1.0-6 DAV00300
C* I10=1 DAV00310
C* I1DMEN=1 DAV00320
C* SET CONVERGENCE TOLERANCE DAV00330
C* DAV00340
C* DELF=1.0-6 DAV00350
C* IF(CRIKEY.EQ.1.AND.SETWGT.EQ.2)DELF=1.0-1 DAV00360
C* IF(CRIKEY.EQ.3)DELF=1.0-5 DAV00370
C* ICNT=0 DAV00380
C* DAV00390
C* INITIALIZE R-MATRIX (XBAR) DAV00400
C* DAV00410
C* IF(BMKEY.LE.0)GO TO 10 DAV00420
C* READ R-MATRIX IN SINGLE PRECISION, THEN STORE IN DOUBLE PRECISION DAV00430
C* DAV00440
C* CALL HMFIL(WKRY,NOFET4,NOFET2,FETVC2,2) DAV00450
C* IK=0 DAV00460
C* DO 5 I=1,NOFET2 DAV00470
C* DO 5 J=1,NOFET4 DAV00480
C* IK=IK+1 DAV00490
C* 5 RMAT(I,J)=WKRY(IK) DAV00500
C* GO TO 20 DAV00510
C* DAV00520
C* INITIALIZE R-MATRIX FROM 'BEST' SET FOUND IN WITHOUT REPLACEMENT P DAV00530
C* DAV00540
C* 10 CALL ORDER(FETVC4,NOFET4) DAV00550
C* DO 15 I=1,NOFET4 DAV00560
C* DO 15 J=1,NOFET2 DAV00570
C* RMAT(I,J)=0.0 DAV00580
C* 15 IF(FETVC4(I).EQ.FETVC2(J))RMAT(I,J)=1.0 DAV00590
C* DAV00600
C* INITIALIZE ROUTINE FOR EVALUATING PARTIALS AND SEP. MEASURE DAV00610
C* DAV00620
C* 20 CALL FINI(COVMTX,AVEMTX,DIVTAB,WEIGHT,S,S2,COVMT2,AVEMT2,PARTLS) DAV00630
C* ITT=0 DAV00640
C* DAV00650
C* COMPUTE PARTIALS FOR INITIAL R-MATRIX DAV00660
C* DAV00670
C* IPART=1 DAV00680
C* CALL FINI2(SMSR,IPART,RMAT,WKRY,IWKSZ,425) DAV00690
C* 25 CONTINUE DAV00700
C* ITT=ITT+1 DAV00710
C* FII=SMSR DAV00720
C* N=NOFET2*NOFET4 DAV00730
C* LEFT=IWKSZ-N*2 DAV00740
C* DAV00750
C* IP=BASE ADDRESS FOR P ARRAY DAV00760
C* IF=BASE ADDRESS FOR H ARRAY (ONLY ONE ROW OF H IN CORE AT A TIME DAV00770
C* IHY=BASE ADDRESS FOR HY ARRAY DAV00780
C* DAV00790

```

FILE: DAVIDN

IP=1	DAV00800
IH=IP+N*2	DAV00810
IHY=IH+N*2	DAV00820
IF1=IHY+N*2	DAV00830
ITEST=IF1+N*2	DAV00840
IF (ITEST.LE.IWRKSZ) GO TO 30	DAV00850
WRITE(6,100) IWRKSZ	DAV00860
CALL CMERR	DAV00870
30 CONTINUE	DAV00880
C* INITIALIZE H AND P ARRAYS - SAVE H AND PARTLS ON SCRATCH FILE	DAV00890
C* ALSO COMPUTE INITIAL OFDK	DAV00900
C* CALL DAVN1(PARTLS,WRKRY(IP),WRKRY(IH))	DAV00910
C* PARTLS STORAGE CAN NOW BE USED IN DAVIDN2 FOR TEMP. STORAGE OF XRA	DAV00920
C* DAVN2 RETURNS A NEW H-MATRIX AND P ARRAY AND CAYMIN	DAV00930
40 CALL DAVN2(BMAT,PARTLS,WRKRY(IP),WRKRY(IH),LEFT,650)	DAV00940
N2=2*N	DAV00950
C* SAVE P ARRAY ON SCRATCH FILE	DAV00960
C* CALL RWRITE(ADRES,WRKRY(IP),N2,LSTAT)	DAV00970
C* WAIT FOR I/O	DAV00980
41 IF (LSTAT.EQ.1) GO TO 41	DAV00990
IF (LSTAT.NE.0) GO TO 70	DAV01000
C* COMPUTE NEW PARTIALS	DAV01010
CALL FINT2(SMSR,IPART,BMAT,WRKRY,IWRKSZ,645)	DAV01020
45 CONTINUE	DAV01030
ITT=ITT+1	DAV01040
ICNT=ICNT+1	DAV01050
SFPMR=SMSR	DAV01060
IF (DARS(SMSR-FII).LT.DELF) GO TO 50	DAV01070
FII=SMSR	DAV01080
IF (ITT.GE.ICOUNT) GO TO 50	DAV01090
SMSR=SFPMSR	DAV01100
C* READ BACK P ARRAY	DAV01110
C* CALL RREAD(ADRES,WRKRY(IP),N2,LSTAT)	DAV01120
46 IF (LSTAT.EQ.1) GO TO 46	DAV01130
IF (LSTAT.NE.0) GO TO 70	DAV01140
C* SURROUTINE DAVN3 UPDATES THE H AND P ARRAYS AND SAVES NEW	DAV01150
C* H AND NEW PARTIALS ON SCRATCH FILE	DAV01160
C* CALL DAVN3(PARTLS,WRKRY(IF1),WRKRY(IP),WRKRY(IH),WRKRY(IHY))	DAV01170
GO TO 40	DAV01180
C* IF CRITERIA = AV. DIVERGENCE - COMPUTE INTERCLASS DIVERGENCES	DAV01190
50 IF (CPIKEY.NE.1) GO TO 60	DAV01200
C* CALL DIVRG1(COVM2,VARSZ4,AVEMT2,DIVTAB,NOCLS2,NOFET4,	DAV01210
WRKRY,IWRKSZ)	DAV01220
60 CONTINUE	DAV01230
C* STORE IN SINGLE PRECISION ARRAY AND	DAV01240
C* WRITE H-MATRIX ON FILE AND PUNCH ON CARDS IF REQUESTED	DAV01250
C*	DAV01260
IK=0	DAV01270
DO 65 I=1,NOFET2	DAV01280
DO 65 J=1,NOFET4	DAV01290
IK=IK+1	DAV01300
WRKRY(IK)=H*AT(J,I)	DAV01310
65 CONTINUE	DAV01320
HMKEY=1	DAV01330
CALL HMFIL(WRKRY,NOFET4,NOFET2,FETVC2,5)	DAV01340
C* IF (CPHKEY.EQ.1) CALL HMFIL(WRKRY,NOFET4,NOFET2,FETVC2,4)	DAV01350
RETURN	DAV01360
70 WRITE(6,200) LSTAT	DAV01370
100 FORMAT(' NOT ENOUGH WORK AREA AVAILABLE IN DAVIDN--IWRKSZ=',I6)	DAV01380
200 FORMAT(' ERROR ON DRUM FILE - SURROUTINE DAVIDN--LSTAT=',I3)	DAV01390
END	DAV01400
	DAV01410
	DAV01420
	DAV01430
	DAV01440
	DAV01450
	DAV01460

ORIGINAL PAGE IS
OF POOR QUALITY

FILE: DIVERG

```

      SUBROUTINE DIVERG(COVMTX,VARSIZ,AVEMTX,DIVTAB,NOCLS,
      NOFET,WRKRY,IWRKSZ)
C*
C* SUBROUTINE TO COMPUTE INTERCLASS DIVERGENCES
C*
      INTEGER VARSIZ
      DOUBLE PRECISION WRKRY(1),T
      DOUBLE PRECISION DIVTAB,DET,TRACE
      DIMENSION COVMTX(VARSIZ,NOCLS), AVEMTX(NOFET,NOCLS),
      DIVTAB(1),T(30)
      INP=1
      GO TO 3
      ENTRY DIVERG(COVMT2,VARSIZ,AVEMT2,DIVTAB,NOCLS,NOFET,WRKRY,IWRKSZ)
      DOUBLE PRECISION COVMT2(VARSIZ,NOCLS),AVEMT2(NOFET,NOCLS)
      INP=0
3 CONTINUE
      ICV1=1
      ICV2=ICV1+VARSIZ
      IW1=ICV2+VARSIZ
      IW2=IW1+VARSIZ
      IF(IWRKSZ/2.GE.IW2+VARSIZ)GO TO 4
      WRITE(6,200)IWRKSZ
      CALL CMERR
4 CONTINUE
      MN=0
      IC=NOCLS-1
      DO 30 I=1,IC
C* FIND INVERSE FOR CLASS I COVAR. MATRIX
      DO 1 IK=1,VARSIZ
      IF(INP.EQ.1)WRKRY(IK)=COVMTX(IK,I)
      IF(INP.EQ.0)WRKRY(IK)=COVMT2(IK,I)
1 CONTINUE
      CALL COLINV(WRKRY(ICV1),NOFET,IERR,3,DET)
      IF(IERR.EQ.0)GO TO 2
      WRITE(6,100) I
      GO TO 30
2 IM=I+1
      DO 20 J=IM,NOCLS
      MN=MN+1
      DO 5 II=1,NOFET
      IF(INP.EQ.1)T(II)=AVEMTX(II,I)-AVEMTX(II,J)
      IF(INP.EQ.0)T(II)=AVEMT2(II,I)-AVEMT2(II,J)
5 CONTINUE
      K=0
      DO 10 II=1,NOFET
      DO 10 JJ=1,II
      K=K+1
      IF(INP.EQ.1)GO TO 6
      WRKRY(ICV2+K-1)=COVMT2(K,J)
      WRKRY(IW1+K-1)=COVMT2(K,J) + T(II)*T(JJ)
      WRKRY(IW2+K-1)=COVMT2(K,I) + T(II)*T(JJ)
      GO TO 10
6 CONTINUE
      WRKRY(ICV2+K-1)=COVMTX(K,J)
      WRKRY(IW1+K-1)=COVMTX(K,J) + T(II)*T(JJ)
      WRKRY(IW2+K-1)=COVMTX(K,I)+T(II)*T(JJ)
10 CONTINUE
C* FIND INVERSE FOR CLASS J COVAR. MATRIX
      CALL COLINV(WRKRY(ICV2),NOFET,IERR,3,DET)
      IF(IERR.EQ.0)GO TO 15
      WRITE(6,100)J
      GO TO 20
15 DIVTAB(MN)=.5*(TRACE(WRKRY(ICV1),WRKRY(IW1),NOFET)
      + TRACE(WRKRY(ICV2),WRKRY(IW2),NOFET)) - NOFET
20 CONTINUE
30 CONTINUE
      RETURN
100 FORMAT(' COVAR FOR CLASS ',I4,' IS NOT POSITIVE DEFINITE')
200 FORMAT(' NOT ENOUGH WORK AREA AVAILABLE IN DIVERG -- IWRKSZ=',I5)
      END

```


FILE: EVALSP

C*	SUBROUTINE EVALSP(SMSR,COVMTX,AVEMTX,S,COVMT2,AVEMT2,S2,	EVA00010
C*	DIVTAB,WEIGHT,IPART,PARTLS,BMAT,WKRY,IWKSZ)	EVA00020
C*	THIS SUBROUTINE COORDINATES THE ROUTINES FOR COMPUTING THE	EVA00030
C*	SEPARABILITY MEASURE FOR A PARTICULAR LINEAR COMBINATION OR	EVA00040
C*	SET OF FEATURES	EVA00050
C*	DOUBLE PRECISION SMSR,COVMT2(1),AVEMT2(1),S2(1),DIVTAB(1)	EVA00060
C*	,PARTLS(1),HMAT(1)	EVA00070
C	INTEGER CRIKEY	EVA00080
C	INCLUDE COMRK1,LIST	EVA00090
C	INCLUDE COMRK7,LIST	EVA00100
	COMMON/INFORM/NOCLS2,NOSUR2,NOFFI2,VARSZ2,TOTVT2,NOFLD2,	EVA00110
	AVAR2,CVVAR2,CLSID2,SURN02,SURH02,FLUSV2,VENTX2,	EVA00120
	FETVC2(30),SUNVC2(75),SURPTH(75),CLSV2(60),	EVA00130
	KFPPTS(60),NOGKP,GKPNAM(60),GRPDEX(61),	EVA00140
	GRPCHK(61),GROUPS(124)	EVA00150
	COMMON/FSL/CFAC,TOTMSR,SEPMSR,PRCKEY,CRIKEY,INCFET,	EVA00160
	INCEVC(30),ICOUNT,SFTWGT,EVALRF(100),FETVC4(30)	EVA00170
	,NOFFI4,VARS74,CORH45,OTAH4,WGHS14,BESTVC(10),DIVSIZ	COM00010
	,STATKY,ADRES0,ADRESF,ADRESF,ADRS1,ADRS2	COM00020
	INTEGER ADRES0,ADRESF,ADRESF,ADRS1,ADRS2,STATKY	COM00030
	DOUBLE PRECISION: CFAC,TOTMSR,SEPMSR	COM00040
CSEND	DIMENSION COVMTX(1), AVEMTX(1), S(1),	COM00050
	2 WEIGHT(1),IPART(1),WKRY(1)	COM00060
	IFULL=0	EVA00250
	GO TO(10,20,30,40),CRIKEY	EVA00260
C*	WEIGHTED AVERAGE DIVERGENCE	EVA00270
	10 CALL AVEDIV(SMSR,COVMTX,S,COVMT2,S2,WKRY,IWKSZ,	EVA00280
	IPART,PARTLS,BMAT,IFULL)	EVA00290
	RETURN	EVA00300
C*	WEIGHTED AVERAGE TRANSFORMED DIVERGENCE	EVA00310
C*	20 CALL TRNDIV(SMSR,COVMTX,AVEMTX,COVMT2,AVEMT2,	EVA00320
C*	WEIGHT,DIVTAB,WKRY,	EVA00330
	IWKSZ,IPART,PARTLS,BMAT,IFULL)	EVA00340
	RETURN	EVA00350
C*	WEIGHTED AVERAGE BHATTACHARYYA DISTANCE	EVA00360
C*	30 CALL BHATCHR(SMSR,COVMTX,AVEMTX,WEIGHT,DIVTAB,	EVA00370
	COVMT2,AVEMT2,WKRY,	EVA00380
	IWKSZ,IPART,PARTLS,BMAT,IFULL)	EVA00390
	RETURN	EVA00400
C*	40 CONTINUE	EVA00410
	RETURN	EVA00420
	END	EVA00430
		EVA00440
		EVA00450
		EVA00460
		EVA00470
		EVA00480
		EVA00490

FILE: EVLFET

	SUBROUTINE EVLFET(COVMTX,AVEMTX,DIVTAB,WEIGHT,COVMT2,AVEMT2, S,S2,WRKRY,IWRKSZ)	EVL00010
C	INCLUDE COMRK1.LIST	EVL00020
C	INCLUDE COMRK7.LIST	EVL00030
	COMMON/INFORM/NOCL S2,NOSUR2,NOFET2,VARSZ2,TOTVT2,NOFLD2, AVAR2,COVAR2,CLSIO2,SURN02,SURD S2,FLDSV2,VERTX2, FETVC2(30),SURVC2(75),SURPTR(75),CLSVC2(60), KFPPTS(60),NOGRP,GRPNAM(60),GRPDEX(61), GRPCHK(41),GROUPS(124)	EVL00040
	COMMON/FSL/CFAC,TOTMSR,SEPMSR,PRCKEY,CRIKEY,INCFET, INCEVC(30),ICOUNT,SETWGT,EVALRF(100),FETVC4(30)	EVL00050
	NOFET4,VARS74,CORHAS,DTAH4,WGHS14,BESTVC(10),DIVSIZ	EVL00060
	STATKY,ADRFSD,ADRESP,ADRESF,ADRSH1,ADRSH2	EVL00070
	INTEGER ADRFSD,ADRESP,ADRESF,ADRSH1,ADRSH2,STATKY	EVL00080
	DOUBLE PRECISION CFAC,TOTMSR,SEPMSR	EVL00090
CSEND	DOUBLE PRECISION COVMT2(1),AVEMT2(1),S2(1),DUM(1),DIVTAB(1)	EVL00100
	INTEGER VARSZ4,FETVC4	EVL00110
	INTEGER CRIKEY	EVL00120
	DIMENSION AVEMTX(1),COVMTX(1), WEIGHT(1), S(1), WRKRY(1)	EVL00130
2	IPART=-1	EVL00140
	CALL GTSTAT(COVMTX,AVEMTX,S,COVMT2,AVEMT2,S2,FETVC4, DUM,WRKRY,IWRKSZ)	EVL00150
	CALL EVALSP(SEPMSR,COVMTX,AVEMTX,S,COVMT2,AVEMT2,S2,DIVTAB, WEIGHT,IPART,DUM,DUM,WRKRY,IWRKSZ)	EVL00160
	IF(CRIKEY.NE.1)RETURN	EVL00170
	CALL DIVRG1(COVMT2,VARSZ4,AVEMT2,DIVTAB,NOCL S2,NOFET4,WRKRY, IWRKSZ)	EVL00180
	RETURN	EVL00190
	END	EVL00200
		EVL00210
		EVL00220
		EVL00230
		EVL00240
		EVL00250
		EVL00260
		EVL00270
		EVL00280
		EVL00290
		EVL00300
		EVL00310

FILE: EXSRCH

```

      SURROUTINE EXSRCH(COVMTX,AVEMTX,DIVTAB,WEIGHT,COVMT2,
      AVEMT2,S,S2,WRKRY,IWRKSZ)
      THIS SURROUTINE USES THE EXHAUSTIVE SEARCH PROCEDURE TO FIND
      THE BEST 'NOFET4' OUT OF 'NOFET2' FEATURES. BY MAXIMIZING THE
      SEPARABILITY MEASURE INDICATED BY 'CRIKEY'.
      INCLUDE COMRK1.LIST
      INCLUDE COMRK7.LIST
      COMMON/INFORM/NOCLS2,NOSUR2,NOFET2,VARSZ2,TOTVT2,NOFLD2,
      AVAR2,COVAR2,CLS102,SURNO2,SURDS2,FLDSV2,VERTX2,
      FFTVC2(30),SUBVC2(75),SURPTR(75),CLSV2(60),
      KFPPTS(60),NOGRP,GHPNAM(60),GRPDEX(61),
      GRPCHK(61),GROUPS(124)
      COMMON/FSL/CFAC,TOTMSR,SEPMSR,PRCKEY,CRIKEY,INCFET,
      INCVEC(30),ICOUNT,SETWGT,EVALBF(100),FETVC4(30),
      NOFET4,VARSZ4,CORBAS,DTAB4,WGHS14,BESTVC(10),DIVSIZ,
      STATKY,ADRES0,ADRESP,ADRESF,ADRSH1,ADRSH2,
      INTEGER ADRES0,ADRESP,ADRESF,ADRSH1,ADRSH2,STATKY
      DOUBLE PRECISION CFAC,TOTMSR,SEPMSR
CSEND
      INTEGER FETVC2,FETVC4,TVEC,VARSZ2,VARSZ4
      INTEGER DIVSIZ,SVFC(30),CRIKEY
      DOUBLE PRECISION TMSR,DIVTAB(DIVSIZ),DUM(1),DM
      DOUBLE PRECISION COVMT2(VARSZ4,1),AVEMT2(NOFET4,1),S2(VARSZ4,1)
      DIMENSION COVMTX(VARSZ2,NOCLS2),
      AVEMTX(NOFET2,NOCLS2),
      S(VARSZ2,NOCLS2)
      DIMENSION TVEC(30)
      DIMENSION WEIGHT(1),WRKRY(1)
C*
C*
C*
      INITIALIZE TVEC FOR GETSET ROUTINE
      DO 1 I=1,NOFET4
      TVFC(I)=I
      TVFC(NOFET4)=NOFET4-1
      SEPMSR=1.E+35
      GET NEXT SET OF FEATURES
      4 CALL GETSET(TVEC,NOFET4,NOFET2,LAST)
      IF(LAST.EQ.0)GO TO 10
C*
C*
C*
      GET SURSET OF STATISTICS FOR THIS SET OF FEATURES
      CALL GTSTAT(COVMTX,AVEMTX,S,COVMT2,AVEMT2,S2,TVEC,DM,WRKRY,IWRKSZ)
C*
C*
C*
      EVALUATE SEPARABILITY MEASURE FOR THIS SET OF FEATURES.
      SET IPART SO PARTIAL DERIVATIVES WILL NOT BE CALCULATED.
      IPART=-1
      CALL EVALSP(TMSR,COVMTX,AVEMTX,S,COVMT2,AVEMT2,S2,DIVTAB,
      WEIGHT,IPART,DUM,DUM,WRKRY,IWRKSZ)
      IF(SEPMSR.LT.TMSR)GO TO 4
      DO 5 I=1,NOFET4
      K=TVFC(I)
      SVFC(I)=K
      FFTVC4(I)=FFTVC2(K)
      SEPMSR=TMSR
      GO TO 4
      5 FINISHED
C*
C*
C*
      COMPUTE INTERCLASS MEASURES FOR FEATURES CHOSEN
      10 CONTINUE
      CALL GTSTAT(COVMTX,AVEMTX,S,COVMT2,AVEMT2,S2,SVFC,DM,WRKRY,IWRKSZ)
      CALL EVALSP(SEPMSR,COVMTX,AVEMTX,S,COVMT2,AVEMT2,S2,DIVTAB,
      WEIGHT,IPART,DUM,DUM,WRKRY,IWRKSZ)
      IF(CRIKEY.NE.1)RETURN
      CALL DIVRG(COVMT2,VARSZ4,AVEMT2,DIVTAB,NOCLS2,
      NOFET4,WRKRY,IWRKSZ)
      20 RETURN
      END

```

EXS00010
 EXS00020
 EXS00030
 EXS00040
 EXS00050
 EXS00060
 EXS00070
 EXS00080
 EXS00090
 EXS00100
 EXS00110
 EXS00120
 EXS00130
 EXS00140
 COM00010
 COM00020
 COM00030
 COM00040
 COM00050
 COM00060
 EXS00220
 EXS00230
 EXS00240
 EXS00250
 EXS00260
 EXS00270
 EXS00280
 EXS00290
 EXS00300
 EXS00310
 EXS00320
 EXS00330
 EXS00340
 EXS00350
 EXS00360
 EXS00370
 EXS00380
 EXS00390
 EXS00400
 EXS00410
 EXS00420
 EXS00430
 EXS00440
 EXS00450
 EXS00460
 EXS00470
 EXS00480
 EXS00490
 EXS00500
 EXS00510
 EXS00520
 EXS00530
 EXS00540
 EXS00550
 EXS00560
 EXS00570
 EXS00580
 EXS00590
 EXS00600
 EXS00610
 EXS00620
 EXS00630
 EXS00640
 EXS00650
 EXS00660
 EXS00670
 EXS00680
 EXS00690
 EXS00700
 EXS00710
 EXS00720
 EXS00730
 EXS00740

1. 2. 3. 4. 5. 6. 7. 8. 9. 10. 11. 12. 13. 14. 15. 16. 17. 18. 19. 20. 21. 22. 23. 24. 25. 26. 27. 28. 29. 30. 31. 32. 33. 34. 35. 36. 37. 38. 39. 40. 41. 42. 43. 44. 45. 46. 47. 48. 49. 50. 51. 52. 53. 54. 55. 56. 57. 58. 59. 60. 61. 62. 63. 64. 65. 66. 67. 68. 69. 70. 71. 72. 73. 74. 75. 76. 77. 78. 79. 80. 81. 82. 83. 84. 85. 86. 87. 88. 89. 90. 91. 92. 93. 94. 95. 96. 97. 98. 99. 100. 101. 102. 103. 104. 105. 106. 107. 108. 109. 110. 111. 112. 113. 114. 115. 116. 117. 118. 119. 120. 121. 122. 123. 124. 125. 126. 127. 128. 129. 130. 131. 132. 133. 134. 135. 136. 137. 138. 139. 140. 141. 142. 143. 144. 145. 146. 147. 148. 149. 150. 151. 152. 153. 154. 155. 156. 157. 158. 159. 160. 161. 162. 163. 164. 165. 166. 167. 168. 169. 170. 171. 172. 173. 174. 175. 176. 177. 178. 179. 180. 181. 182. 183. 184. 185. 186. 187. 188. 189. 190. 191. 192. 193. 194. 195. 196. 197. 198. 199. 200. 201. 202. 203. 204. 205. 206. 207. 208. 209. 210. 211. 212. 213. 214. 215. 216. 217. 218. 219. 220. 221. 222. 223. 224. 225. 226. 227. 228. 229. 230. 231. 232. 233. 234. 235. 236. 237. 238. 239. 240. 241. 242. 243. 244. 245. 246. 247. 248. 249. 250. 251. 252. 253. 254. 255. 256. 257. 258. 259. 260. 261. 262. 263. 264. 265. 266. 267. 268. 269. 270. 271. 272. 273. 274. 275. 276. 277. 278. 279. 280. 281. 282. 283. 284. 285. 286. 287. 288. 289. 290. 291. 292. 293. 294. 295. 296. 297. 298. 299. 300. 301. 302. 303. 304. 305. 306. 307. 308. 309. 310. 311. 312. 313. 314. 315. 316. 317. 318. 319. 320. 321. 322. 323. 324. 325. 326. 327. 328. 329. 330. 331. 332. 333. 334. 335. 336. 337. 338. 339. 340. 341. 342. 343. 344. 345. 346. 347. 348. 349. 350. 351. 352. 353. 354. 355. 356. 357. 358. 359. 360. 361. 362. 363. 364. 365. 366. 367. 368. 369. 370. 371. 372. 373. 374. 375. 376. 377. 378. 379. 380. 381. 382. 383. 384. 385. 386. 387. 388. 389. 390. 391. 392. 393. 394. 395. 396. 397. 398. 399. 400. 401. 402. 403. 404. 405. 406. 407. 408. 409. 410. 411. 412. 413. 414. 415. 416. 417. 418. 419. 420. 421. 422. 423. 424. 425. 426. 427. 428. 429. 430. 431. 432. 433. 434. 435. 436. 437. 438. 439. 440. 441. 442. 443. 444. 445. 446. 447. 448. 449. 450. 451. 452. 453. 454. 455. 456. 457. 458. 459. 460. 461. 462. 463. 464. 465. 466. 467. 468. 469. 470. 471. 472. 473. 474. 475. 476. 477. 478. 479. 480. 481. 482. 483. 484. 485. 486. 487. 488. 489. 490. 491. 492. 493. 494. 495. 496. 497. 498. 499. 500. 501. 502. 503. 504. 505. 506. 507. 508. 509. 510. 511. 512. 513. 514. 515. 516. 517. 518. 519. 520. 521. 522. 523. 524. 525. 526. 527. 528. 529. 530. 531. 532. 533. 534. 535. 536. 537. 538. 539. 540. 541. 542. 543. 544. 545. 546. 547. 548. 549. 550. 551. 552. 553. 554. 555. 556. 557. 558. 559. 560. 561. 562. 563. 564. 565. 566. 567. 568. 569. 570. 571. 572. 573. 574. 575. 576. 577. 578. 579. 580. 581. 582. 583. 584. 585. 586. 587. 588. 589. 590. 591. 592. 593. 594. 595. 596. 597. 598. 599. 600. 601. 602. 603. 604. 605. 606. 607. 608. 609. 610. 611. 612. 613. 614. 615. 616. 617. 618. 619. 620. 621. 622. 623. 624. 625. 626. 627. 628. 629. 630. 631. 632. 633. 634. 635. 636. 637. 638. 639. 640. 641. 642. 643. 644. 645. 646. 647. 648. 649. 650. 651. 652. 653. 654. 655. 656. 657. 658. 659. 660. 661. 662. 663. 664. 665. 666. 667. 668. 669. 670. 671. 672. 673. 674. 675. 676. 677. 678. 679. 680. 681. 682. 683. 684. 685. 686. 687. 688. 689. 690. 691. 692. 693. 694. 695. 696. 697. 698. 699. 700. 701. 702. 703. 704. 705. 706. 707. 708. 709. 710. 711. 712. 713. 714. 715. 716. 717. 718. 719. 720. 721. 722. 723. 724. 725. 726. 727. 728. 729. 730. 731. 732. 733. 734. 735. 736. 737. 738. 739. 740. 741. 742. 743. 744. 745. 746. 747. 748. 749. 750. 751. 752. 753. 754. 755. 756. 757. 758. 759. 760. 761. 762. 763. 764. 765. 766. 767. 768. 769. 770. 771. 772. 773. 774. 775. 776. 777. 778. 779. 780. 781. 782. 783. 784. 785. 786. 787. 788. 789. 790. 791. 792. 793. 794. 795. 796. 797. 798. 799. 800. 801. 802. 803. 804. 805. 806. 807. 808. 809. 810. 811. 812. 813. 814. 815. 816. 817. 818. 819. 820. 821. 822. 823. 824. 825. 826. 827. 828. 829. 830. 831. 832. 833. 834. 835. 836. 837. 838. 839. 840.

CSEND

```
SPSR - SEPARABILITY MEASURE RETURNED FROM EVALSP
TPART- TRIGGER TO COMPUTE PARTIALS OR NOT
      - LESS THAN 0 - COMPUTE PARTIALS
      - GREATER THAN 0 - DO NOT COMPUTE PARTIALS
RMTAT - H-MATRIX
```

[illegible]

ORIGINAL PAGE IS
OF POOR QUALITY

FILE: FINT1

| | | |
|-----|--|----------|
| C* | WRKRY- WORKING STORAGE | F1N00800 |
| C* | IWRKZ- SIZE OF WRKRY IN COMPUTER WORDS | F1N00810 |
| C* | | F1N00820 |
| | DIMENSION BMAT(1),WRKRY(1) | F1N00830 |
| | DOUBLE PRECISION BMAT,SMR | F1N00840 |
| C* | | F1N00850 |
| C* | GET TRANSFORMED STATISTICS FOR THIS B-MATRIX | F1N00860 |
| C* | | F1N00870 |
| | CALL GTSTAT(COVMTX,AVEMTX,S,COVMT2,AVEMT2,S2,DUM,BMAT,WRKRY, | F1N00880 |
| | IWRKSZ) | F1N00890 |
| | CALL EVALSP(SPSR,COVMTX,AVEMTX,S,COVMT2,AVEMT2,S2,DIVTAB, | F1N00900 |
| | WEIGHT,IPART,PARTLS,BMAT,WRKRY,IWRKSZ) | F1N00910 |
| | SMR=DARS(SPSR) | F1N00920 |
| | ITT=ITT+1 | F1N00930 |
| | ICNT=ICNT+1 | F1N00940 |
| | IF(ICYCLE.LE.1)GO TO 50 | F1N00950 |
| | IF(ICNT.LE.30)GO TO 50 | F1N00960 |
| | WRITE(4,700) | F1N00970 |
| | ICNT=0 | F1N00980 |
| | RETURN 1 | F1N00990 |
| 50 | CONTINUE | F1N01000 |
| | IF(IPART.LT.0)RETURN | F1N01010 |
| | ICNT=0 | F1N01020 |
| | ICYCLE=ICYCLE+1 | F1N01030 |
| | IF(CRIKEY.EQ.1)RATIO=SMR/TOTMSR | F1N01040 |
| | IF(CRIKEY.NE.1)RATIO=TOTMSR/SMR | F1N01050 |
| | WRITE(4,600)ICYCLE,ITT,SMR,RATIO | F1N01060 |
| | RETURN | F1N01070 |
| 100 | FORMAT(///25X,'CONVERGENCE CHARACTERISTIC SUMMARY FOR THE DAVIDON-LETCHER-POWELL PROCEDURE',/25X,76(' ')/) | F1N01080 |
| | * LETCHER-POWELL PROCEDURE | F1N01090 |
| 200 | FORMAT(//35X,'NUMBER OF LINEAR COMBINATIONS',10X,'=',I10/ | F1N01100 |
| | * 35X,'DESIGN NO. OF FUNCTIONAL EVAL.',8X,'=',I10) | F1N01110 |
| 300 | FORMAT(35X,'MAX. WEIGHTED AVERAGE DIVERGENCE (D)',3X,'=',E10.5) | F1N01120 |
| 400 | FORMAT(35X,'MIN. WEIGHTED AV. TRANS. DIVERGENCE (T)',3X,'=',E10.5) | F1N01130 |
| 500 | FORMAT(35X,'MIN. WEIGHTED AV. BHATTACHARYA DIS. (H)',3X,'=',E10.5) | F1N01140 |
| 600 | FORMAT(20X,I4,T37,I4,T61,F12.7,T90,E12.7) | F1N01150 |
| 700 | FORMAT(' MAX. ITERATIONS PER CYCLE - BEGIN NEW CYCLE') | F1N01160 |
| | END | F1N01170 |
| | | F1N01180 |

FILE: GENRPT

```

SURROUTINE GENRPT(CLSNAM,WEIGHT,DIVTAB,WRKRY,IWRKSZ,FETVEC)
C DIMENSION CLSNAM(NOCLS2),WEIGHT(1)
INCLUDE COMRK6.LIST
DOUBLE PRECISION RATIO
DOUBLE PRECISION WRKRY(1),DIVTAB(1)
C DIMENSION FETVEC(30)
C INCLUDE COMRK1.LIST
C INCLUDE COMRK7.LIST
COMMON/INFORM/NOCLS2,NOSUR2,NOFET2,VARSZ2,TOTVT2,NOFLD2,
* AVAR2,COVAR2,CLS1D2,SURNO2,SURDS2,FLOSV2,VERTX2,
* FETVC2(30),SUHVC2(75),SURPTR(75),CLSV2(60),
* KEPPTS(60),NOGRP,GRPNAM(60),GRPDEX(61),
* GRPCHK(61),GROUPS(124)
COMMON/GLORAL/HEAD(63),MAPTAP,DATAPE,SAVTAP,BMFILE,BMKEY,
* HISFIL,HISKEY,TRFORM,ERIPTR,ERPKEY,MAPUNT,NOFILF,
* DRIMAD,DRMWDOS,PAGSIZ,DATFIL,STAFIL,ASAV,ASAVFL
* ,NHSTIN,NHSTFI,SCTRUN,MAPFIL
* ,DOTUNT,DOTFIL,NCHPAS,TRNSFL,BMTRFL,HISTFL,PCHUNT,
* CRDUNT,PRUNT,RANDIO
COMMON/FSL/CFAC,TOTMSR,SEPMSR,PRCKEY,CRIKEY,INCFET,
* INCVEC(30),ICOUNT,SETWGT,EVALBF(100),FETVC4(30)
* ,NOFET4,VARS74,CORBAS,DAR4,WGHS14,RESTVC(10),DIVSIZ
* ,STATKY,ADRESO,ADRESP,ADRESF,ADRS1,ADRS2,ADRS3
INTEGER ADPFSO,ADRESP,ADRESF,ADRS1,ADRS2,STATKY
DOUBLE PRECISION CFAC,TOTMSR,SEPMSR
C$END
INTEGER CRIKEY,PRCKEY
DIMENSION PRC(6,6),CRI(13,4)
DATA PRC/EXHA,USTI,VF S,EARC,H , , ,
* WITH,OUT,REPL,ACEM,ENT , , ,
* DAVI,DON,FLET,CHER,-POW,ELL ,
* EVAL,B-MATR,IX R,EQUE,ST ,
* EVAL,FE,ATUR,ES R,EQUE,ST ,
* PFST,K , OF,N , PA,SSES/
DATA CRI/WEIG,HTED,AVE,PRAGE,DIV,ERGE,NCE ,
* ,WEIG,HTED,AV,TRA,NSFO,RMED,DIV,
* ,ERGE,NCE,AV,TRA,NSFO,RMED,DIV,
* ,WEIG,HTED,AV,BHA,TTAC,HARY,YA D,
* ,ISTA,NCE,AV,BHA,TTAC,HARY,YA D,
* ,PROB,ABIL,ITY,OF M,ISCL,ASSI,FICA,
* ,TION,OF,TRAI,NING,DAT,A /
INTEGER DIVSIZ,PAGSIZ
INTEGER FETVC4
IPCNT=27
WRITE(6,HEAD)
WRITE(6,100)(PRC(I,PRCKEY),I=1,6),(CRI(I,CRIKEY),I=1,13)
IR=1
IE=15
5 IF(IF.GT.NOFET2)IF=NOFET2
WRITE(6,105)(FETVC2(I),I=IR,IE)
IF(IF.EQ.NOFET2)GO TO 10
IR=IE+1
IE=IF+15
GO TO 5
10 CONTINUE
WRITE(6,125)
IF(PRCKEY.EQ.3.OR.PRCKEY.EQ.4)WRITE(6,110)NOFET4
IF(PRCKEY.EQ.3.OR.PRCKEY.EQ.4)GO TO 14
IR=1
IE=14
12 IF(IF.GT.NOFET4)IF=NOFET4
IF(PRCKEY.LT.3)WRITE(6,120)(FETVC4(I),I=IR,IE)
IF(PRCKEY.GE.5)WRITE(6,115)(FETVEC(I),I=IR,IE)
IF(IE.EQ.NOFET4)GO TO 18
IR=IE+1
IE=IF+14
GO TO 12
14 CONTINUE
IF(CRIKEY.EQ.1)SEPMSR=DAHS(SEPMSR)
IF(PRCKEY.EQ.3)WRITE(6,130)SEPMSR
IF(PRCKEY.EQ.4)WRITE(6,130)SEPMSR
IF(PRCKEY.GE.5)WRITE(6,145)SEPMSR
IF(PRCKEY.LT.3)WRITE(6,140)SEPMSR
IF(CRIKEY.EQ.1)RATIO=SEPMSR/TOTMSR
IF(CRIKEY.NE.1)RATIO=TOTMSR/SEPMSR
IF(CRIKEY.NE.1)WRITE(6,150)TOTMSR,RATIO
IF(CRIKEY.EQ.1)WRITE(6,155)TOTMSR,RATIO
WRITE(6,160)

```

GEN00010
 GEN00020
 GEN00030
 GEN00040
 GEN00050
 GEN00060
 GEN00070
 GEN00080
 GEN00090
 GEN00100
 GEN00110
 GEN00120
 GEN00130
 GEN00140
 GEN00150
 GEN00160
 GEN00170
 GEN00180
 GEN00190
 GEN00200
 GEN00210
 GEN00220
 GEN00230
 GEN00240
 GEN00250
 GEN00260
 GEN00270
 GEN00280
 GEN00290
 GEN00300
 GEN00310
 GEN00320
 GEN00330
 GEN00340
 GEN00350
 GEN00360
 GEN00370
 GEN00380
 GEN00390
 GEN00400
 GEN00410
 GEN00420
 GEN00430
 GEN00440
 GEN00450
 GEN00460
 GEN00470
 GEN00480
 GEN00490
 GEN00500
 GEN00510
 GEN00520
 GEN00530
 GEN00540
 GEN00550
 GEN00560
 GEN00570
 GEN00580
 GEN00590
 GEN00600
 GEN00610
 GEN00620
 GEN00630
 GEN00640
 GEN00650
 GEN00660
 GEN00670
 GEN00680
 GEN00690
 GEN00700
 GEN00710
 GEN00720
 GEN00730
 GEN00740
 GEN00750
 GEN00760
 GEN00770
 GEN00780
 GEN00790

FILE: GENRPT

```

      IF (CRIKEY.EQ.1) WRITE(6,170)
      IF (CRIKEY.EQ.2) WRITE(6,175)
      IF (CRIKEY.EQ.3) WRITE(6,180)
      IF (PRCKEY.LE.2) WRITE(6,190)
      IF (PRCKEY.EQ.3) WRITE(6,200)
      IF (PRCKEY.EQ.4) WRITE(6,200)
      IF (PRCKEY.GE.5) WRITE(6,205)
      WRITE(6,125)
C*
C* READ INTERCLASS MEASURE FOR ALL FEATURES - COMPUTED AND SAVED IN
C* PRELIM.
      NW=DIVSIZ*2
      CALL WREAD(ADRESO,WRKRY,NW,ISTAT)
19  IF(ISTAT.EQ.1) GO TO 19
      NC=NOCLS2-1
      IK=0
      DO 20 I=1,NC
      K=I+1
      DO 20 J=K,NOCLS2
      IK=IK+1
      IF (CRIKEY.EQ.1) RATIO=DIVTAB(IK)/WRKRY(IK)
      IF (CRIKEY.NE.1) RATIO=WRKRY(IK)/DIVTAB(IK)
      WRITE(6,210) CLSNAM(I),CLSNAM(J),
      * WEIGHT(IK),DIVTAB(IK),WRKRY(IK),RATIO
      IPCNT=IPCNT+1
      IF (IPCNT.LT.PAGSIZ) GO TO 20
      IPCNT=17
      IF (IK.EQ.DIVSIZ) GO TO 20
      WRITE(6,HEAD)
      WRITE(6,160)
      IF (CRIKEY.EQ.1) WRITE(6,170)
      IF (CRIKEY.EQ.2) WRITE(6,175)
      IF (CRIKEY.EQ.3) WRITE(6,180)
      IF (PRCKEY.LE.2) WRITE(6,190)
      IF (PRCKEY.EQ.3) WRITE(6,200)
      IF (PRCKEY.EQ.4) WRITE(6,200)
      IF (PRCKEY.GE.5) WRITE(6,205)
      WRITE(6,125)
20  CONTINUE
      IF (CRIKEY.NE.2) RETURN
C*
C* GET DIVERGENCE BACK FOR PLOTS
C*
      DO 30 I=1,DIVSIZ
      WRKRY(I)=-14.*DLOG(WRKRY(I))
      DIVTAB(I)=-14.*DLOG(DIVTAB(I))
30  CONTINUE
100 FORMAT(1X,T35,'RESULTS FOR CHANNEL SELECTION ACTIVITY USING:/'
      * 1X,T40,'OPTIMIZATION PROCEDURE - ',A4/'
      * 1X,T40,'SEPARABILITY MEASURE - ',A4/'
105 FORMAT(1X,T40,'CHANNELS CONSIDERED - ',I2,'/')
110 FORMAT(1X,T35,'NO. OF LINEAR COMBINATIONS - ',I3)
115 FORMAT(1X,T35,'CHANNELS EVALUATED - ',I3)
      * 16(I2,'/')
120 FORMAT(1X,T35,'CHANNELS SELECTED - ',I3)
      * 12,I5(' ','I2'))
125 FORMAT(1X,T35,'SEPARABILITY MEASURE FOR LINEAR COMB. - ',F14,8)
130 FORMAT(1X,T35,'SEPARABILITY MEASURE FOR SELECTED CHANNELS - ',F14,8)
140 FORMAT(1X,T35,'SEPARABILITY MEASURE FOR EVALUATE REQUEST - ',F14,8)
145 FORMAT(1X,T35,'MINIMUM SEP.MEASURE (USING ALL CHANNELS) - ',F14,8)
150 FORMAT(1X,T35,'MAXIMUM SEP.MEASURE (USING ALL CHANNELS) - ',F14,8)
155 FORMAT(1X,T35,'RATIO',T78,'-',F14,8)
160 FORMAT(1X,T50,'INTERCLASS SEPARABILITY TABLE' //)
170 FORMAT(1X,T73,'INTERCLASS DIVERGENCE')
175 FORMAT(1X,T65,'INTERCLASS TRANSFORMED DIVERGENCE')
180 FORMAT(1X,T65,'INTERCLASS WHATTACHARYYA DISTANCE')
190 FORMAT(1X,T22,'SUBCLASS PAIR',T41,'WEIGHT',T58,'SELECTED CHANNELS'
      * T40,'ALL CHANNELS',T99,'RATIO')
200 FORMAT(1X,T22,'SUBCLASS PAIR',T41,'WEIGHT',T58,'LINEAR COMBINATION'
      * T40,'ALL CHANNELS',T99,'RATIO')
205 FORMAT(1X,T24,'SUBCLASS PAIR',T41,'WEIGHT',T58,'EVALUATED CHANNELS'
      * T40,'ALL CHANNELS',T99,'RATIO')
210 FORMAT(20X,A4,7X,A4,4X,E8.3,T60,D14,8,T78,D14,8,T96,D14,8)

```

GEN00800
 GEN00810
 GEN00820
 GEN00830
 GEN00840
 GEN00850
 GEN00860
 GEN00870
 GEN00880
 GEN00890
 GEN00900
 GEN00910
 GEN00920
 GEN00930
 GEN00940
 GEN00950
 GEN00960
 GEN00970
 GEN00980
 GEN00990
 GEN01000
 GEN01010
 GEN01020
 GEN01030
 GEN01040
 GEN01050
 GEN01060
 GEN01070
 GEN01080
 GEN01090
 GEN01100
 GEN01110
 GEN01120
 GEN01130
 GEN01140
 GEN01150
 GEN01160
 GEN01170
 GEN01180
 GEN01190
 GEN01200
 GEN01210
 GEN01220
 GEN01230
 GEN01240
 GEN01250
 GEN01260
 GEN01270
 GEN01280
 GEN01290
 GEN01300
 GEN01310
 GEN01320
 GEN01330
 GEN01340
 GEN01350
 GEN01360
 GEN01370
 GEN01380
 GEN01390
 GEN01400
 GEN01410
 GEN01420
 GEN01430
 GEN01440
 GEN01450
 GEN01460
 GEN01470
 GEN01480
 GEN01490
 GEN01500
 GEN01510
 GEN01520
 GEN01530
 GEN01540
 GEN01550
 GEN01560
 GEN01570
 GEN01580

FILE: GENRPT

RETURN
END

GEN01590
GEN01600

~~10-34~~

104

FILF: GETSET

```

      SUBROUTINE GETSET(VEC,NOFET4,NOFET2,LAST)
      INTEGER VEC
C*
C*  SUBROUTINE TO RETURN A UNIQUE SET OF INDICES FOR NOFET4 FEATURES
C*  IN VEC.  VEC SHOULD BE INITIALIZED TO 1,2,3...(NOFET4-1) BEFORE
C*  ENTERING THIS ROUTINE FOR THE FIRST TIME.
C*
      DIMENSION VEC(1)
      DO 30 JJ=1,NOFET4
      J2=NOFET4-JJ+1
      J3=J2+1
      VEC(J2) = VEC(J2)+1
      IF (J3.GT.NOFET4) GO TO 20
      DO 10 JJJ=J3,NOFET4
10  VEC(JJJ)=VEC(JJJ-1)+1
20  IF (VEC(NOFET4).LE.NOFET2) GO TO 40
30  CONTINUE
      LAST=0
      RETURN
40  LAST=1
      RETURN
      END
```

FILE: GTSTAT

```

      SUBROUTINE GTSTAT(COVMTX,AVEMTX,S,COVMT2,AVEMT2,S2,
      VEC,HMAT,WRKRY,IWRKSZ)
      THIS SUBROUTINE SELECTS THE SUBSETS OF THE STATISTICAL
      PARAMETERS COVMTX,AVEMTX AND S DEFINED BY VEC OR HMAT AND
      STORES THE SUBSETS INTO COVMT2,AVEMT2, AND S2 RESPECTIVELY.
      INCLUDE COMHKL.LIST
      INCLUDE COMHKT.LIST
      COMMON/INFOWM/NOCLS2,NOSUR2,NOFET2,VARSZ2,TOTVT2,NOFLD2,
      AVAR2,COVAR2,CLSTD2,SUMNO2,SUMDS2,FLDSV2,VERTX2,
      FETVC2(30),SUMVC2(75),SUMPTH(75),CLSV2(60),
      KEPTS(60),NOGRP,GRPNAM(60),GRPDEX(61),
      GRPCHK(61),GROUPS(124)
      COMMON/FSL/CFAC,TOTMSR,SEPMSS,PRCKEY,CRIKEY,INCFET,
      INCVEC(30),ICOUNT,SETWGT,EVALRF(100),FETVC4(30),
      NOFET4,VARSZ4,CUMHAS,DIARA,WGMS14,RESTVC(10),DIVSIZ
      ,STATKY,ADHESD,ADRESP,ADRESF,ADRSH1,ADRSH2
      INTEGER ADHESD,ADRESP,ADRESF,ADRSH1,ADRSH2,STATKY
      DOUBLE PRECISION CFAC,TOTMSR,SEPMSS
      CSEND
      INTEGER PRCKEY,CRIKEY,VEC,VARSZ2,VARSZ4
      DOUBLE PRECISION COVMT2(VARSZ4,1),AVEMT2(NOFET4,1),
      S2(VARSZ4,1),HMAT(NOFET4,1),WRKRY(1),SUM
      DIMENSION COVMTX(VARSZ2,NOCLS2),
      AVEMTX(NOFET2,NOCLS2),
      S(VARSZ2,NOCLS2),VEC(1)
      IF WITHOUT REPLACEMENT OR EX. SEARCH PROCEDURE, SELECT ELEMENTS
      DETERMINED BY 'VEC'.
      GO TO (5,5,20,20,5,5),PRCKEY
      5 CONTINUE
      DO 10 I=1,NOCLS2
      IK=0
      DO 10 J=1,NOFET4
      K=VEC(J)
      LOC=K*(K-1)/2
      AVEMT2(J,1)=AVEMTX(K,1)
      DO 10 L=1,J
      IK=IK+1
      IW=VEC(L)*LOC
      COVMT2(IK,1)=COVMTX(IW,1)
      IF(CRIKEY.NE.1)GO TO 10
      S2(IK,1)=S(IW,1)
      10 CONTINUE
      RETURN
      DAVIDON OR USER INPUT PROCEDURE, MULTIPLY R-MATRIX
      20 CONTINUE
      IW1=1
      ITFST=IW1 + NOFET2*NOFET4
      IF(IWRKSZ/2.GE.ITFST)GO TO 30
      WRITE(6,100)IWRKSZ
      CALL CMEMW
      30 CONTINUE
      DO 40 I=1,NOCLS2
      DO 40 J=1,NOFET4
      SUM=0.0
      DO 40 K=1,NOFET2
      SUM=SUM+AVEMTX(K,1)*HMAT(J,K)
      40 AVEMT2(J,1)=SUM
      40 CONTINUE
      CALL TRANSR(COVMTX,COVMT2,WRKRY(IW1),HMAT)
      IF(CRIKEY.EQ.1)CALL TRANSR(S,S2,WRKRY(IW1),HMAT)
      RETURN
      100 FORMAT(' NOT ENOUGH WORK AREA IN GTSTAT -- IWRKSZ=',I5)
      END

```

FILE: MT1

C*
C*
C*
C*
C*

MT1 - MATRIX B IS STORED IN SYMMETRIC NOTATION AND SINGLE PRE.

C*

```
SUBROUTINE MT1(A,R,C,M,N)
  DOUBLE PRECISION A(M,N),C(M,N),SUM
  DIMENSION M(1)
  MATRIX R IS STORED IN SYMMETRIC NOTATION
  DO 60 II=1,M
    DO 60 JJ=1,N
      SUM=0.0
      DO 55 KK=1,M
        IF(KK.GE.JJ) IK=KK*(KK-1)/2 + JJ
        IF(KK.LT.JJ) IK=JJ*(JJ-1)/2 + KK
55 SUM = SUM + A(II,KK)*R(IK)
  60 C(II,JJ)=SUM
  RETURN
END
```

FILE: MT2

```
C*  SURROUTINE TO FORM PRODUCT OF MATRICES A*B AND STORE IN C.
C*
C*  MT2 - MATRIX A IS STORED IN SYMMETRIC NOTATION
C*
      SURROUTINE MT2(A,R,C,M,N)
      DOUBLE PRECISION A(1),R(M,N),C(M,N),SUM1
      DO 90 II=1,M
      DO 90 JJ=1,N
      SUM1=0.0
      DO 85 IJ=1,M
      IF(IJ.GE.II) IK=IJ*(IJ-1)/2+II
      IF(IJ.LE.II) IK=II*(II-1)/2+IJ
      AS  SUM1=SUM1 + A(IK)*R(IJ,JJ)
      90  C(II,JJ)=SUM1
      RETURN
      END
```

FILE: MT3

```
C*
C*  MATRICES A AND/OR B MAY BE STORED IN SYMMETRIC NOTATION
C*  L=M IF A IS SYMMETRIC ---- N=M IF B IS SYMMETRIC
SURROUTINE MT3(A,B,C,L,M,N,ISYMA,ISYMB)
DOUBLE PRECISION A(1),B(1),C(L,N),SUM
DO 70 II=1,L
DO 70 JJ=1,N
SUM=0.0
DO 65 KK=1,M
IF (ISYMA.EQ.1) GO TO 61
IK=L*(KK-1) + II
GO TO 62
41 IF (KK.GE.II) IK=KK*(KK-1)/2 + II
IF (KK.LT.II) IK=II*(II-1)/2 + KK
62 IF (ISYMB.EQ.1) GO TO 63
JK=M*(JJ-1) + KK
GO TO 65
63 IF (KK.GE.JJ) JK=KK*(KK-1)/2 + JJ
IF (KK.LT.JJ) JK=JJ*(JJ-1)/2 + KK
65 SUM=SUM + A(IK)*B(JK)
70 C(II,JJ)=SUM
RETURN
END
```

FILE: MT4

```
C*
C*      SUPROUTINE MT4(A,R,C,L,M,N,ISYM)
C*
C*      MATRIX A CAN BE STORED FULL OR IN SYMMETRIC NOTATION
C*      ISYM=1  IF A IS SYMMETRIC
C*      ISYM=0  IF A IS FULL
C*
C*      DOUBLE PRECISION A(1),B(M,N),C(L,N),SUM
      DO 95 II=1,L
      DO 95 JJ=1,N
      SUM=0.0
      DO 99 IJ=1,M
      IF (ISYM.EQ.1) GO TO 85
      IK=L*(IJ-1) + II
      GO TO 99
85  IF (IJ.GE.II) IK=IJ*(IJ-1)/2 + II
      IF (IJ.LT.II) IK=II*(II-1)/2 + IJ
      SUM = SUM + A(IK)*B(IJ,JJ)
95  C(II,JJ)=SUM
      RETURN
      END
```

FILE: PLOT

```

C ***** SURROUTINE PLOT(X,Y,NOX,MAXX,ILABX,ILABY,ICODE,IOPT) ***** PLO00010
C THIS SUBROUTINE WAS WRITTEN BY J.K.DALY OF TRW FOR THE ASTEP PLO00020
C PROGRAM. IT WAS MODIFIED SLIGHTLY FOR USE IN THIS PROGRAM BY PLO00030
C R. MINTER PLO00040
C PLO00050
C PLO00060
C PLO00070
C PLO00080
C PLO00090
C PLO00100
C PLO00110
C PLO00120
C PLO00130
C PLO00140
C PLO00150
C PLO00160
C PLO00170
C PLO00180
C PLO00190
C PLO00200
C PLO00210
C PLO00220
C PLO00230
C PLO00240
C PLO00250
C PLO00260
C PLO00270
C PLO00280
C PLO00290
C PLO00300
C PLO00310
C PLO00320
C PLO00330
C PLO00340
C PLO00350
C PLO00360
C PLO00370
C PLO00380
C PLO00390
C PLO00400
C PLO00410
C PLO00420
C PLO00430
C PLO00440
C PLO00450
C PLO00460
C PLO00470
C PLO00480
C PLO00490
C PLO00500
C PLO00510
C PLO00520
C PLO00530
C PLO00540
C PLO00550
C PLO00560
C PLO00570
C PLO00580
C PLO00590
C PLO00600
C PLO00610
C PLO00620
C PLO00630
C PLO00640
C PLO00650
C PLO00660
C PLO00670
C PLO00680
C PLO00690
C PLO00700
C PLO00710
C PLO00720
C PLO00730
C PLO00740
C PLO00750
C PLO00760
C PLO00770
C PLO00780
C PLO00790

C *****
C THIS SUBROUTINE WAS WRITTEN BY J.K.DALY OF TRW FOR THE ASTEP
C PROGRAM. IT WAS MODIFIED SLIGHTLY FOR USE IN THIS PROGRAM BY
C R. MINTER
C
C --ICODE AND IOPT ARE SET TO 0, AND ILABX,ILABY ARE BLANK.
C WRITE STATEMENTS WERE ADDED FOR LABELING.
C *****
C PROGRAMMER - J.K. DALY
C DATE - FEBRUARY, 1973
C MODIFIED FOR IRM 370-148 R HANSEN,C HORTON DEC,1977
C INPUT VARIABLES
C X - ARRAY CONTAINING X COORDINATES (TYPE REAL)
C Y - ARRAY CONTAINING Y COORDINATES (TYPE REAL)
C NOX - NO. OF X COORDINATES INPUT (MAX. 39)
C MAXX - MAXIMUM VALUE OF POINT ON SCALE TO BE INPUT (INTEGER)
C ILABX - LABEL OF THE X-AXIS (MAX. OF 78 CHARACTERS ALLOWED)
C ILABY - LABEL OF THE Y-AXIS (MAX. OF 78 CHARACTERS ALLOWED)
C ICODE - 1 = RUN IS MADE ON TSS,INVALID 12-77
C 0 = RUN IS MADE ON 1108
C IOPT - 0 = 45 DEGREE ANGLE LINE TO BE DRAWN ON PLOT
C 1 = NO 45 DEG ANGLE LINE,INVALID 12-77
C INTERNAL VARIABLES
C FMDX - WHICH WORD OF THE FORMAT ARRAY(FMTARY) TO START
C LOADING THE LABELS.
C FMTARY - ARRAY TO BE BUILT FOR EACH LINE OF THE PLOT
C I - INDEX FOR INNER LOOP FOR DETERMINING THE LOCATION
C OF THE POINT ON THE X-AXIS
C INDEX - MAXIMUM NO. OF WORDS OF FMTARY TO BE WRITTEN (TSS=11,
C 1108=15)
C INITLN - INITIAL LINE TO BEGIN PRINTING THE LABELS
C INUM - NUMBER OF POINTS TO FALL IN SAME PLACE ON PLOT.
C J - INDEX FOR LOOP TO FIND NEXT HIGHEST Y-COORDINATE.
C K - INDEX FOR OUTER LOOP BUILDING AND PRINTING EACH
C LINE OF PLOT.
C LABNDX - INDEX FOR LABARY
C LASTLN - LAST LINE TO PRINT LABEL ON PLOT
C M - INDEX OF SCLARY
C MHORIZ - NO. OF CHARACTERS/PLOT
C MLNCT - NO. OF LINES/PLOT
C NDX - INDEX TO 'WICLAR'
C NOXPT - NO. OF POINTS TO FALL BETWEEN LABELED POINTS
C ON X-AXIS
C NOYPT - NO. OF POINTS TO FALL BETWEEN LABELED POINTS
C ON Y-AXIS
C SCLARY - ARRAY CONTAINING POINTS TO BE PRINTED AS LABELS
C ON THE X- AND Y-AXIS
C WICLAR - ARRAY CONTAINING LINE NO.S OF BOTH KINDS OF PLOTS.
C DEPENDING ON THE LINE NO. DETERMINES WHICH LABEL
C WILL BE PRINTED.
C XLNVLU - VALUE OF EACH POINT ON THE X-AXIS. USED TO CALCULATE
C EACH POINT OF XSCLAR (X SCALE ARRAY) IN SURROUTINE
C SCALF.
C XSCLAR - ACTUAL VALUE OF EACH POINT ON X-AXIS (TYPE REAL)
C YLNVLU - VALUE OF EACH POINT ON Y-AXIS. USED TO CALCULATE EACH
C POINT OF YSCLAR (Y SCALE ARRAY)
C *****
C INCLUDE COMMRK6.LIST
C INCLUDE COMMRK7.LIST
C COMMON/GLOBAL/HEAD(63),MAPTAP,DATAPE,SAVTAP,RMFILE,RMKEY,
C HISFIL,HISKEY,TRFORM,ERIPTP,ERPKEY,MAPUNT,NOFILF,
C DRUMAD,DRMADS,PAGSIZ,DATFIL,STAFIL,ASAV,ASAVFL
C NHSTUN,NHSTFI,STRUN,MAPFIL
C DOTUNT,DOTFIL,NCHPAS,TPNSFL,BMTRFL,HISTFL,PCHUNT,
C CRDUNT,PRUNT,KANDIO
C COMMON/FSL/CFAC,TOTMSR,SEPMSR,PRCKEY,CHRIKEY,INCFET,
C INCVEC(30),ICOUNT,SFTWGT,FVALHF(100),FFTVC4(30)
C ,NOFET4,VARSZ4,CORHAS,DTA94,WGHS14,HESTVC(10),DIVSI7
C ,STATKY,ADHESD,ADRESP,ADRESF,ADRSH1,ADRSH2
C INTEGER ADRESD,ADRESP,ADRESF,ADRSH1,ADRSH2,STATKY
C INTEGER PCDA,PCDX,PCDLK,PCDSTR
C DOUBLE PRECISION CFAC,TOTMSR,SEPMSR
C *****
C SEND
C DIMENSION FMTARY(23)

```

FILE: PLOT

```

REAL
DOUBLE PRECISION X(NOX),Y(NOX)      YSCLAR(57),XSCLAR(85)
REAL SCLARY(N)
INTEGER CRKEY,PRCKEY
INTEGER LABARY(40),LARNOX,WICLAR(10),FMDX
INTEGER ILAPLX(20),ILABLY(20),FMTARY,ASTRIC,0
C
C LOGICAL*1 FMTA(92),LTEST(4),LSTAR,LOG1
C
DATA LOG1/ZF1/,LRLANK/Z40/,LSTAR/Z5C/
DATA FMTARY/23*4H /,ASTRIC/4H****/,IBLANK/4H /
DATA IPOINT/4H 1/
DATA RCDRLK/Z40/,RCDSTH/Z5C/,HCD9/ZF9/,BCDX/ZE7/
DATA WICLAR /55.54,52.50,49.42,41.40,39.38/
C
EQUIVALENCE (FMTA(1),FMTARY(1)),(LTEST,LTEST(1))
C
C BLANK X AND Y LABELS
DO 100 I=1,20
  ILAPLX(I)=IBLANK
  ILABLY(I)=IBLANK
C
C* FIND MAXX
MAXX=1
AMAX=0.
IF(CRKEY.EQ.3)GO TO 900
IKCT=0
C
C COUNT NUMBER OF X(I) GREATER THAN 700. SET AMAX = LARGEST X(I)
C SET MAXX=700 UNLESS MORE THAN 1/5 OF ELEMS GRTR 70, THEN MAXX=1400
DO 800 I=1,NOX
  IF(X(I).GT.700)IKCT=IKCT+1
  800 CONTINUE
  MAXX=700
  IF(FLOAT(IKCT)/FLOAT(NOX) .GT. .2)MAXX=1400
  900 CONTINUE
C
C WRITE MESSAGE 'SEPARABILITY TO BE GAINED MAP'
CALL SETMRG(66,2,64)
WRITE(6,1695)
1695 FORMAT(50X,'SEPARABILITY-TO-BE-GAINED MAP')
C
C BLANK LABARY
DO 1000 II=1,40
  LABARY(II) = IBLANK
1000 CONTINUE
C
C BLANK FMTARY
DO 1001 II=1,23
  1001 FMTARY(II) = IBLANK
C
C STORE X- AND Y-AXIS LABELS INTO LABEL ARRAY(LABARY)
DO 1020 II=1,20
  IF (ILABLY(II).EQ.0) GO TO 1020
  LABARY(II) = ILABLY(II)
  IF (ILAPLX(II).EQ.0) GO TO 1020
  LABARY(II+20) = ILAPLX(II)
1020 CONTINUE
C
C* * * SET VARIABLES FOR PRINT PLOT ON 1108
LARNOX = 1
N = 7
MLNCT = 56
MPORTZ = 45
NOYPT = 12
NOYPT = 4
NOSPAC = 44
INITLN = 55
LASTLN = 44
NOX = 1
FMDX = 3
INDEX = 23
C
C CALL SCALE TO FOR SCALE FOR PLOT
1050 CONTINUE
C
C

```

PL000800
 PL000810
 PL000820
 PL000830
 PL000840
 PL000850
 PL000860
 PL000870
 PL000880
 PL000890
 PL000900
 PL000910
 PL000920
 PL000930
 PL000940
 PL000950
 PL000960
 PL000970
 PL000980
 PL000990
 PL001000
 PL001010
 PL001020
 PL001030
 PL001040
 PL001050
 PL001060
 PL001070
 PL001080
 PL001090
 PL001100
 PL001110
 PL001120
 PL001130
 PL001140
 PL001150
 PL001160
 PL001170
 PL001180
 PL001190
 PL001200
 PL001210
 PL001220
 PL001230
 PL001240
 PL001250
 PL001260
 PL001270
 PL001280
 PL001290
 PL001300
 PL001310
 PL001320
 PL001330
 PL001340
 PL001350
 PL001360
 PL001370
 PL001380
 PL001390
 PL001400
 PL001410
 PL001420
 PL001430
 PL001440
 PL001450
 PL001460
 PL001470
 PL001480
 PL001490
 PL001500
 PL001510
 PL001520
 PL001530
 PL001540
 PL001550
 PL001560
 PL001570
 PL001580

FILE: PLOT

```

      CALL SCALE(MAXX,MLNCT,INCR,YSCLAR,XSCLAR,SCLARY,XLNVLU,YLNVLU,
      *      MHORIZ,NOXPT,NOYPT)
      XLNVLU = XLNVLU/2.0
      YLNVLU = YLNVLU/2.0
1000 CONTINUE
C
C
C FIND Y COORDINATE
DO 1400 I1400 = 1,MLNCT
  K = MLNCT - I1400 + 1
  DO 1100 J=1,NOX
    YY = Y(J)
    IF(YY.GT.YSCLAR(K))GO TO 1180
    IF(YY.LE.YSCLAR(K-1))GO TO 1180
    IF(YY.GT.FLOAT(MAXX)) GO TO 1180
1100 CONTINUE
C
C * * * Y COORDINATE FOUND. NOW FIND POS. OF X COOD.
DO 1160 I1160 = 1,MHORIZ
  I = MHORIZ - I1160 + 1
C CHECK FOR X VALUE IN RANGE
  XX = X(I)
  IF(XX.GT.XSCLAR(I)) GO TO 1160
  IF(XX.LE.XSCLAR(I-1)) GO TO 1160
  IF(XX.GT.FLOAT(MAXX)) GO TO 1180
C MOVE ITH CHAR TO ITEST, RIGHT JUSTIFIED, 0 FILLED
  ITEST = 0
  LTEST(4) = FMTA(I)
C CHECK FOR BLANK OR STAR IMPLYING NO PREVIOUS OCCURANCES
  IF (ITEST.EQ.RCDRLK .OR. ITEST.EQ.RCDSTR) GO TO 1140
C CHECK FOR X IMPLYING MAXIMUM COUNT ALREADY REACHED
  IF (ITEST.EQ.RCDX) GO TO 1180
C ADD 1 TO COUNT. IF MORE THAN 9 OCCURANCES, CHG COUNT TO X
  ITEST = ITEST + 1
  IF (ITEST.GT.RCD9) ITEST =RCDX
C STORE UPDATED COUNT BACK IN FMT ARRAY
  FMTA(I) = LTEST(4)
  GO TO 1180
C BLANK OR STAR, STORE 1 FOR FIRST OCCURANCE
1140 FMTA(I) = LOG1
  GO TO 1180
1160 CONTINUE
1180 CONTINUE
C
C
C BUILD 45 DEGREE ANGLE LINE OF PLOT
IF (MOD(K,2).EQ.0) GO TO 1220
NOSPAC = NOSPAC - 3
LTEST(4) = FMTA(NOSPAC)
IF (ITEST.NE.LHLANK) GO TO 1220
FMTA(NOSPAC) = LSTAR
C
C
C SET UP LABEL
1220 CONTINUE
C DETERMINE WHETHER OR NOT LABEL IS PRINTED
IF (K.LT.LASTLN) GO TO 1340
IF (K.GT.INITLN) GO TO 1340
IF (K.NE.WICLAR(NDX)) GO TO 1340
IF (K.EQ.52.OR.K.EQ.40) GO TO 1320
C
C MOVE LABARY TO FMTARY
IJ = FMDX + 5
IF (MOD(K,2).EQ.0) IJ =FMDX + 5
DO 1240 II=FMDX,IJ
  FMTARY(II) = LABARY(LABNDX)
  LABNDX = LABNDX + 1
1240 CONTINUE
  NDX = NDX + 1
  GO TO 1340
1320 CONTINUE
  FMTARY(FMDX+3)=IHLANK
  NDX = NDX + 1
1340 CONTINUE
  IF (K.EQ.MLNCT) GO TO 1380
  IF (MOD(K+7,8).EQ.0) GO TO 1380
  WRITE (6,1560)FMTARY
  GO TO 1440

```

PL001590
 PL001600
 PL001610
 PL001620
 PL001630
 PL001640
 PL001650
 PL001660
 PL001670
 PL001680
 PL001690
 PL001700
 PL001710
 PL001720
 PL001730
 PL001740
 PL001750
 PL001760
 PL001770
 PL001780
 PL001790
 PL001800
 PL001810
 PL001820
 PL001830
 PL001840
 PL001850
 PL001860
 PL001870
 PL001880
 PL001890
 PL001900
 PL001910
 PL001920
 PL001930
 PL001940
 PL001950
 PL001960
 PL001970
 PL001980
 PL001990
 PL002000
 PL002010
 PL002020
 PL002030
 PL002040
 PL002050
 PL002060
 PL002070
 PL002080
 PL002090
 PL002100
 PL002110
 PL002120
 PL002130
 PL002140
 PL002150
 PL002160
 PL002170
 PL002180
 PL002190
 PL002200
 PL002210
 PL002220
 PL002230
 PL002240
 PL002250
 PL002260
 PL002270
 PL002280
 PL002290
 PL002300
 PL002310
 PL002320
 PL002330
 PL002340
 PL002350
 PL002360
 PL002370

FILE: PLOT

```
1560 FORMAT (1H ,22X,1H*,23A4)
1380 CONTINUE
C PRINT PLOT FOR 1108
IF (M.LE.1) GO TO 1480
IF (K.FQ.MLNCT) WRITE (6,1660)FMTARY
1660 FORMAT(1H ,21X,2H *,23A4)
IF (K.FQ.MLNCT) GO TO 1440
WRITE (6,1540)SCLARY(M),FMTARY
1540 FORMAT(1H ,14X,E7.2,2H *,23A4)
M = M - 1
1440 CONTINUE
C
C WRITE X AND Y LABELS AS SELECTED
DO 1460 I1460=1,INDEX
FMTARY(I1460)=IBLANK
1460 CONTINUE
IF (K.NF.40)GO TO 1030
IF (PRCKEY.EQ.3 .OR. PRCKEY.EQ.4)WRITE(6,1710)
1710 FORMAT(1H*,*LINEAR*)
IF (PRCKEY.EQ.1 .OR. PRCKEY.EQ.2)WRITE(6,1720)
1720 FORMAT(1H*,*SELECTED*)
1030 IF (K.NF.38)GO TO 1035
IF (PRCKEY.EQ.3 .OR. PRCKEY.EQ.4) WRITE(6,1730)
1730 FORMAT(1H*,*COMB.*)
IF (PRCKEY.EQ.1 .OR. PRCKEY.EQ.2) WRITE(6,1740)
1740 FORMAT(1H*,*CHANNELS*)
1035 CONTINUE
1480 CONTINUE
C
C * * * PRINT X-AXIS LINE AND LABELS
C * * * 1108 SIZE PLOT
WRITE (6,1580)(SCLARY(L),L=1,8)
1580 FORMAT(1H ,22X,7(12H* * * * * ),1H*/15X,E7.2,3X,7(5X,E7.2))
WRITE(6,1750)
1750 FORMAT(60X,*ALL CHANNELS*)
IF (CRIKEY.EQ.3)WRITE(6,1690)
1690 FORMAT(75X,*INTERCLASS BHATTACHARYYA DISTANCE*)
IF (CRIKEY.NF.3)WRITE(6,1700)
1700 FORMAT(75X,*PAIRWISE DIVERGENCE*)
RETURN
END
```

PL002380
PL002390
PL002400
PL002410
PL002420
PL002430
PL002440
PL002450
PL002460
PL002470
PL002480
PL002490
PL002500
PL002510
PL002520
PL002530
PL002540
PL002550
PL002560
PL002570
PL002580
PL002590
PL002600
PL002610
PL002620
PL002630
PL002640
PL002650
PL002660
PL002670
PL002680
PL002690
PL002700
PL002710
PL002720
PL002730
PL002740
PL002750
PL002760
PL002770
PL002780
PL002790
PL002800

ORIGINAL PAGE IS
OF POOR QUALITY

FILE: PRELIM

```

C* SURROUTINE PRELIM(COVMTX,AVEMTX,DIVTAB,WEIGHT,S,
C* WRKRY,WRKSIZ)
C*
C* THIS SUBROUTINE PERFORMS SOME OF THE PRELIMINARY TASKS FOR
C* FEATURE SELECTION. THE INTERCLASS MEASURES USING ALL FEATURES
C* ARE COMPUTED AND STORED ON A SCRATCH FILE FOR LATER PRINTING.
C*
C* IN ADDITION, THIS SUBROUTINE COMPUTES THE 'S' MATRICES USED IN
C* COMPUTING WEIGHTED AVERAGE DIVERGENCE IF CRIKEY=1. IF WEIGHTS
C* ARE TO BE SET BY DEFAULT, THE SUBROUTINE ALSO PERFORMS THIS TASK.
C*
C* IMPLICIT INTEGER(A-Z)
C* INCLUDE COMPK1.LIST
C* INCLUDE COMPK7.LIST
C* COMMON/INFORM/NOCLS2,NOSUR2,NOFET2,VARSZ2,TOTVT2,NOFLD2,
C* AVAR2,COVAR2,CLSID2,SURN02,SURDS2,FLDSV2,VERTX2,
C* FETVC2(30),SUHVC2(75),SURPTR(75),CLSV2(60),
C* KPPPTS(60),NOGRP,GRPNAM(60),GRPDEX(61),
C* GRPCHK(61),GROUPS(124)
C* COMMON/FSL/CFAC,TOTMSR,SEPMSR,PRCKEY,CRIKEY,INCFET,
C* INCVEC(30),ICOUNT,SETWGT,EVALRF(100),FETVC4(30),
C* NOFET4,VARSZ4,CORBAS,DTAH4,WGHS14,RESTVC(10),DIVSIZ
C* ,STATKY,ADRESO,ADHESP,ADRESF,ADRSH1,ADRSH2
C* INTEGER ADRESO,ADRESF,ADRESH1,ADRSH2,STATKY
C* DOUBLE PRECISION CFAC,TOTMSR,SEPMSR
C*
C* COMMON/HESTKN/ KPPPTS(60), IPRIOR, KREST, NCPASS
C* REAL COVMTX(VARSZ2,NOCLS2), AVEMTX(NOFE2,NOCLS2),
C* WEIGHT(DIVSIZ), S(VARSZ2,NOCLS2)
C* REAL T(30),APRWGT(780),ANUMPX
C* DOUBLE PRECISION DIVTAB(DIVSIZ),DUM
C* DOUBLE PRECISION WRKRY(1)
C* REAL TW(60)
C* ANUMPX = 0.
C* DO 1 I=1,NOCLS2
C* ANUMPX = ANUMPX + FLOAT(KPPPTS(I))
C* K = 0
C* NC = NOCLS2 - 1
C* DO 2 I=1,NC
C* IK = I + 1
C* DO 2 II = IK,NOCLS2
C* K = K + 1
C* APRWGT(K) = FLOAT(KPPPTS(I)*KPPPTS(II))/(ANUMPX**2)
C* APRWGT(K) = SQRT(APRWGT(K))
C* IF (IPRIOR.NE.0) WRITE(6,990)
C* 990 FORMAT(1H1,' A PRIORI WEIGHT MULTIPLIERS AND TOTAL NO PIXELS')
C* IF (IPRIOR.NE.0) WRITE(6,1000) (APRWGT(I),I=1,DIVSIZ),ANUMPX
C* 1000 FORMAT(1H1,' A PRIORI WEIGHT MULTIPLIERS AND TOTAL NO PIXELS')
C* IF (SETWGT.EQ.2) GO TO 6
C* DO 5 I=1,DIVSIZ
C* WEIGHT(I) = 1.0
C* IF (IPRIOR.EQ.0) GO TO 9
C* DO 7 I=1,DIVSIZ
C* WEIGHT(I) = WEIGHT(I)*APRWGT(I)
C* 9 CONTINUE
C* SET IPART SO PARTIALS WILL NOT BE COMPUTED.
C* IPART=-1
C* IFILL=1
C* GO TO(10,70,80,90),CRIKEY
C*
C* CRITERIA - WEIGHTED AVERAGE DIVERGENCE
C* --COMPUTE INTERCLASS DIVERGENCES
C* --SET WEIGHTS IF SETWGT=0
C* --COMPUTE S-MATRICES
C* --COMPUTE WEIGHTED AVERAGE DIVERGENCE FOR ALL FEATURES
C*
C* 10 CALL DIVERG(COVMTX,VARSZ2,AVEMTX,DIVTAB,NOCLS2,NOFET2,
C* WRKRY,WRKSIZ)
C* IF (SETWGT.NE.0) GO TO 25
C* DO 20 K=1,DIVSIZ
C* WEIGHT(K) = DEXP(-DIVTAB(K)/16.)
C* IF (IPRIOR.NE.0) WEIGHT(K) = WEIGHT(K)*APRWGT(K)
C* 20 COMPUTE S-MATRICES
C* 25 CONTINUE
C* DO 30 J=1,NOCLS2
C* DO 30 I=1,VARSZ2
C* S(I,J)=0.0
C* 30 NC=NOCLS2-1
C* DO 60 N=1,NOCLS2

```

PRE00010
 PRE00020
 PRE00030
 PRE00040
 PRE00050
 PRE00060
 PRE00070
 PRE00080
 PRE00090
 PRE00100
 PRE00110
 PRE00120
 PRE00130
 PRE00140
 PRE00150
 PRE00160
 PRE00170
 PRE00180
 PRE00190
 COM00010
 COM00020
 COM00030
 COM00040
 COM00050
 COM00060
 PRE00270
 PRE00280
 PRE00290
 PRE00300
 PRE00310
 PRE00320
 PRE00330
 PRE00340
 PRE00350
 PRE00360
 PRE00370
 PRE00380
 PRE00390
 PRE00400
 PRE00410
 PRE00420
 PRE00430
 PRE00440
 PRE00450
 PRE00460
 PRE00470
 PRE00480
 PRE00490
 PRE00500
 PRE00510
 PRE00520
 PRE00530
 PRE00540
 PRE00550
 PRE00560
 PRE00570
 PRE00580
 PRE00590
 PRE00600
 PRE00610
 PRE00620
 PRE00630
 PRE00640
 PRE00650
 PRE00660
 PRE00670
 PRE00680
 PRE00690
 PRE00700
 PRE00710
 PRE00720
 PRE00730
 PRE00740
 PRE00750
 PRE00760
 PRE00770
 PRE00780
 PRE00790
 PRE00800

FILE: PRELIM

```
C*
C* SELECT ALL WEIGHTS FOR CLASS N
C*
KT=0
K=0
MN=0
DO 35 J=1,NC
  IJ=J+1
  DO 35 I=IJ,NOCLS2
    K=K+1
    IF (J.NE.N.AND.I.NE.N) GO TO 35
    KT=KT+1
    TW(KT)=WEIGHT(K)
35 CONTINUE
DO 50 M=1,NOCLS2
  IF (M.EQ.N) GO TO 50
  MN=MN+1
  DO 40 I=1,NOFET2
    T(I)=AVEMTX(I,N)-AVEMTX(I,M)
40 J=0
    DO 45 I=1,NOFET2
      DO 45 K=1,I
        J=J+1
45 S(J,N)=S(J,N)+TW(MN)*(COVMTX(J,M)+T(I)*T(K))
50 CONTINUE
60 CONTINUE
C* COMPUTE CFAC
CFAC=0
DO 65 I=1,DIVSIZ
  CFAC=CFAC+WEIGHT(I)
CFAC=1./CFAC
C* COMPUTE AVERAGE WEIGHTED DIVERGENCE
CALL AVEDIV(TOTMSR,COVMTX,S,DUM,DUM,WRKRY,WRKSIZ,
  IPART,DUM,DUM,IFULL)
TOTMSR=DABS(TOTMSR)
GO TO 85
C*
C* CRITERIA - WEIGHTED AVERAGE TRANSFORMED DIVERGENCE
C*
70 CALL TRNDIV(TOTMSR,COVMTX,AVEMTX,DUM,DUM,WEIGHT,DIVTAB,
  WRKRY,WRKSIZ,IPART,DUM,DUM,IFULL)
GO TO 85
C*
C* CRITERIA - BHATTACHARYYA DISTANCE
C*
80 CALL BHTCHR(TOTMSR,COVMTX,AVEMTX,WEIGHT,DIVTAB,DUM,DUM,
  WRKRY,WRKSIZ,IPART,DUM,DUM,IFULL)
C* SAVE INTERCLASS WEIGHTS ON DRUM
85 IQ=DIVSIZ*2
CALL KWRITE(ADRES0,DIVTAB,IQ,ISTAT)
86 IF (ISTAT.EQ.1) GO TO 86
90 RETURN
END
```

PRE00810
PRE00820
PRE00830
PRE00840
PRE00850
PRE00860
PRE00870
PRE00880
PRE00890
PRE00900
PRE00910
PRE00920
PRE00930
PRE00940
PRE00950
PRE00960
PRE00970
PRE00980
PRE00990
PRE01000
PRE01010
PRE01020
PRE01030
PRE01040
PRE01050
PRE01060
PRE01070
PRE01080
PRE01090
PRE01100
PRE01110
PRE01120
PRE01130
PRE01140
PRE01150
PRE01160
PRE01170
PRE01180
PRE01190
PRE01200
PRE01210
PRE01220
PRE01230
PRE01240
PRE01250
PRE01260
PRE01270
PRE01280
PRE01290
PRE01300
PRE01310
PRE01320
PRE01330
PRE01340

ORIGINAL PAGE IS
OF POOR QUALITY

FILE: PRTFLD

```

C      SUPROUTINE PRTFLD(COVMTX,AVEMTX,FLDMTX,VERTEX,
C      CLSNAM,SUBNAM)
C
C      IMPLICIT INTEGER (A-H,O-Z)
C      PRINT TRAINING FIELDS AND CLASS STATISTICS
C
C      INCLUDE COMMK1.LIST
C      DATA LPRN/1CH(1/,RPRN/1)/
C      INCLUDE COMMK6.LIST
C      INCLUDE COMMK7.LIST
C      COMMON/INFURN/NOCLS2,NOSUR2,NOFET2,VARSZ2,TOTVT2,NOFLD2,
C      AVAR2,COVAR2,CLS1D2,SURNO2,SURDS2,FLDSV2,VERTX2,
C      FETVC2(30),SURVC2(75),SURPTR(75),CLSV2(60),
C      KEPPTS(60),NOGRP,GHPNAM(60),GRPDEX(61),
C      GRPCHK(61),GROUPS(124)
C      COMMON/GLOBAL/HEAD(63),MARTAP,DATAP,SAVTAP,RMFILE,RMKEY,
C      HISFIL,HISKEY,TRFORM,ERIPTR,ERPKEY,MAPUNT,NOFILE,
C      DRUMAD,DRMWDOS,PAGSIZ,DATFIL,STAFIL,ASAV,ASAVFL
C      NMSTUN,NMSTFI,STRUN,MAPEIL
C      DOTUNT,DOTFIL,NCHPAS,TRNSFL,BMTRFL,HISTFL,PCHUNT,
C      CRDUNT,PRTUNT,RANDIO
C      COMMON/FSL/CFAC,TOTMSR,SEPMSR,PCKEY,CRIKEY,INCFET,
C      INCVEC(10),ICOUNT,SETWGT,EVALHF(100),FETVC4(30)
C      NOFET4,VARSZ4,CORHAS,DAT4,WGHS14,RESTVC(10),DIVSIZ
C      STATKY,ADRESU,ADRESP,ADRESF,ADRS1,ADRS2,ADRS2,STATKY
C      INTEGER ADRESU,ADRESP,ADRESF,ADRS1,ADRS2,STATKY
C      DOUBLE PRECISION CFAC,TOTMSR,SEPMSR
CSEND
C      DIMENSION COVMTX(VARSZ2,NOSUB2),AVEMTX(NOFET2,NOSUB2),
C      FLDMTX(4,NOFLD2),VERTEX(2,TOTVT2),
C      CLSNAM(NOCLS2),SUBNAM(NOSUB2)
C      DATA ONE/1/,SCFSZ2/3600/,RCDTWO/12/
C
C      WRITE OUT TRAINING FIELDS
C      -----
C      CALL WRTFLD(FLDMTX,VERTEX,NOFLD2,1,CLSNAM,SUBNAM)
C
C      PRINT THE COVARIANCE AND MEAN
C      -----
C      210 IF (STATKY.F0.0) GO TO 300
C      CNT = 7*(5+3+2*NOFET2)*((NOFET2+11)/12)
C      CNT = PAGSIZ/CNT
C      INC = CNT
C
C      DO 290 ICLAS = 1,NOSUB2
C      IF (INC.LT.CNT) GO TO 220
C      WRITE (6,HEAD)
C      INC = 0
C      220 WRITE(6,230) SUBNAM(ICLAS)
C      230 FORMAT(/1X,'SUBCLASS ',A4)
C      DO 240 LOC=1,NOFET2+12
C      STOP = LOC+11
C      IF (STOP.GT. NOFET2) STOP = NOFET2
C      WRITE(6,250) (LPRN,FETVC2(I),RPRN,I=LOC,STOP)
C      240 WRITE (6,260) (AVEMTX(I,ICLAS),I=LOC,STOP)
C      250 FORMAT(10X,12(A3,I2,A1,3X))
C      260 FORMAT('0MEAN',3X,12F9.2)
C      WRITE (6,240)
C      240 FORMAT(10 COVARIANCE MATRIX)
C      CALL WRTMTX(COVMTX(1,ICLAS),NOFET2,RCDTWO)
C      INC = INC+1
C      290 CONTINUE
C
C      300 CONTINUE
C      RETURN
C      END

```

PRT00010
 PRT00020
 PRT00030
 PRT00040
 PRT00050
 PRT00060
 PRT00070
 PRT00080
 PRT00090
 PRT00100
 PRT00110
 PRT00120
 PRT00130
 PRT00140
 PRT00150
 PRT00160
 PRT00170
 PRT00180
 PRT00190
 PRT00200
 PRT00210
 PRT00220
 PRT00230
 PRT00240
 COM00010
 COM00020
 COM00030
 COM00040
 COM00050
 COM00060
 PRT00320
 PRT00330
 PRT00340
 PRT00350
 PRT00360
 PRT00370
 PRT00380
 PRT00390
 PRT00400
 PRT00410
 PRT00420
 PRT00430
 PRT00440
 PRT00450
 PRT00460
 PRT00470
 PRT00480
 PRT00490
 PRT00500
 PRT00510
 PRT00520
 PRT00530
 PRT00540
 PRT00550
 PRT00560
 PRT00570
 PRT00580
 PRT00590
 PRT00600
 PRT00610
 PRT00620
 PRT00630
 PRT00640
 PRT00650
 PRT00660
 PRT00670
 PRT00680
 PRT00690
 PRT00700
 PRT00710
 PRT00720
 PRT00730
 PRT00740
 PRT00750

FILE: SCALE

```

      SURROUTINE SCALE(MAXX,MLNCT,INCR,YSCALAR,XSCALAR,SCLARY,XLNVLU,
      • YLNVLU,MHORIZ,NOXPT,NOYPT)
C
      INCLUDE COMRK7.LIST
      COMMON/FSL/CFAC,TOTMSR,SEPMSR,PRCKEY,CRIKEY,INCFET,
      • INCVFC(30),ICOUNT,SETWGT,EVALRF(100),FFTV4(30)
      • NOFET4,VAR574,CORHAS,DTAH4,WGHS14,RESTVC(10),DIVSIZ
      • STATKY,ADRES0,ADRESP,ADRESF,ADRSH1,ADPSH2
      INTEGER ADRES0,ADRESP,ADRESF,ADRSH1,ADRSH2,STATKY
      DOUBLE PRECISION CFAC,TOTMSR,SEPMSR
CSEND
      INTEGER CRIKEY
      REAL YSCALAR(MLNCT), XSCALAR(MHORIZ),XLNVLU,YLNVLU,RSAVE,XPTS,YPTS
      DIMENSION SCLARY(R)
C * * * ZF40 OUT SCALE LABEL ARRAY - SCLARY
      DO 1000 I=1,M
      SCLARY(I) = 0
      1000 CONTINUE
      IF(CRIKEY.EQ.3 .OR. CRIKEY.EQ.4)GO TO 1070
C * * * DETERMINE LABELS FOR X AND Y AXIS
      INCR = MAXX/7
      IF (MOD(INCR,5) .NE. 0) INCR = 5 - MOD(INCR,5) + INCR
      ISAVE = INCR
      DO 1020 I=2,R
      SCLARY(I) = ISAVE
      ISAVE = INCR + ISAVE
      1020 CONTINUE
C * * * DETERMINE THE VALUE OF EACH POINT ON THE X AND Y AXIS
      XPTS = NOXPT
      YPTS = NOYPT
      RSAVE = INCR
      XLNVLU = RSAVE/XPTS
      YLNVLU = RSAVE/YPTS
      YSCALAR(1) = 0.0
      XSCALAR(1) = 0.0
      RSAVE = YLNVLU
      DO 1040 I=2,MLNCT
      YSCALAR(I) = RSAVE
      RSAVE = RSAVE + YLNVLU
      1040 CONTINUE
      RSAVE = XLNVLU
      DO 1060 I=2,MHORIZ
      XSCALAR(I) = RSAVE
      RSAVE = RSAVE + XLNVLU
      1060 CONTINUE
      RETURN
      1070 CONTINUE
      XPTS=NOXPT
      YPTS=NOYPT
      IEXP=7
      DO 1075 I=1,R
      SCLARY(I) = 1. / 10.**IEXP
      1075 IEXP = IEXP-1
      K=0
      DO 1080 I=1,7
      K=K+1
      RNCRE = (SCLARY(I+1) - SCLARY(I))/YPTS
      YSCALAR(K)=SCLARY(I)
      DO 1090 J=2,NOYPT
      K=K+1
      1080 YSCALAR(K)=YSCALAR(K-1) + RNCRE
      K=0
      DO 1090 I=1,7
      K=K+1
      XSCALAR(K)=SCLARY(I)
      RNCRE = (SCLARY(I+1)-SCLARY(I))/XPTS
      DO 1090 J=2,NOXPT
      K=K+1
      1090 XSCALAR(K)=XSCALAR(K-1)+RNCRE
      1090 CONTINUE
      RETURN
      END

```

SCA00010
 SCA00020
 SCA00030
 SCA00040
 SCA00050
 SCA00060
 SCA00070
 SCA00080
 SCA00090
 SCA00100
 SCA00110
 SCA00120
 SCA00130
 SCA00140
 SCA00150
 SCA00160
 SCA00170
 SCA00180
 SCA00190
 SCA00200
 SCA00210
 SCA00220
 SCA00230
 SCA00240
 SCA00250
 SCA00260
 SCA00270
 SCA00280
 SCA00290
 SCA00300
 SCA00310
 SCA00320
 SCA00330
 SCA00340
 SCA00350
 SCA00360
 SCA00370
 SCA00380
 SCA00390
 SCA00400
 SCA00410
 SCA00420
 SCA00430
 SCA00440
 SCA00450
 SCA00460
 SCA00470
 SCA00480
 SCA00490
 SCA00500
 SCA00510
 SCA00520
 SCA00530
 SCA00540
 SCA00550
 SCA00560
 SCA00570
 SCA00580
 SCA00590
 SCA00600
 SCA00610
 SCA00620
 SCA00630
 SCA00640
 SCA00650
 SCA00660
 SCA00670
 SCA00680
 SCA00690
 SCA00700
 SCA00710

C

IMPLICIT INTEGER (A-H,O-Z)

RETURNS.. SUPERVISOR INFORMATION AND REDUCED STATISTICS

2

C

```
COMMON/ESTKN/ KPPPTS(40), IPRIOR, KBEST, NCPASS
DIMENSION PROC(3,4), CH1(3,3)
DATA CH1/ DIVE, URGEN, CE, TRAN, S, D, IV, BHAT, T, D,
*IST, /
DATA PROC/ EX, SEA, RCH, WITH, OUT, RPLC,
*DAVI, ION, EVL, B MA, TRIX, EVAL, UATE, /
*TR, I, X, PASS, /
DIMENSION ARRAY(1), CARD(42)
DIMENSION SUBRAY(1)
REAL SUBRAY
DATA CHCD/ C, /, SB CD/ S, /, PH CD/ P, /, UB CD/ U, /, FB CD/ F, /
DATA HB CD/ R, /
```

NOGRP=0

C

SF T00010
 SF T00020
 SF T00030
 SF T00040
 SF T00050
 SF T00060
 SF T00070
 SF T00080
 SF T00090
 SF T00100
 SF T00110
 SF T00120
 SF T00130
 SF T00140
 SF T00150
 SF T00160
 SF T00170
 SF T00180
 SF T00190
 SF T00200
 SF T00210
 SF T00220
 SF T00230
 SF T00240
 SF T00250
 SF T00260
 SF T00270
 SF T00280
 SF T00290
 SF T00300
 SF T00310
 SF T00320
 SF T00330
 SF T00340
 SF T00350
 SF T00360
 SF T00370
 SF T00380
 SF T00390
 SF T00400
 SF T00410
 SF T00420
 SF T00430
 SF T00440
 SF T00450
 SF T00460
 SF T00470
 SF T00480
 SF T00490
 SF T00500
 SF T00510
 SF T00520
 SF T00530
 SF T00540
 SF T00550
 SF T00560
 SF T00570
 SF T00580
 SF T00590
 SF T00600
 SF T00610
 SF T00620
 SF T00630
 SF T00640
 SF T00650
 SF T00660
 SF T00670
 SF T00680
 SF T00690
 SF T00700
 SF T00710
 SF T00720
 SF T00730
 SF T00740
 SF T00750
 SF T00760

[illegible]

FILE SETUP4

| | | | |
|-----|------|---|----------|
| C* | 170 | J=NUMBER(CARD,COL,NUMVEC,0) | SET02290 |
| | | ICOUNT=NUMVEC(1) | SET02300 |
| | | GO TO 10 | SET02310 |
| C* | | DATE CARD | SET02320 |
| C* | 180 | READ(30,6000)DATE | SET02330 |
| | | REWIND RRUNIT | SET02340 |
| | | GO TO 10 | SET02350 |
| C* | | HED1 CARD | SET02360 |
| C* | 190 | READ(30,6000)HED1 | SET02370 |
| | | REWIND RRUNIT | SET02380 |
| | | GO TO 10 | SET02390 |
| C* | | HED2 CARD | SET02400 |
| C* | 200 | READ(30,6000)HED2 | SET02410 |
| | | REWIND RRUNIT | SET02420 |
| | | GO TO 10 | SET02430 |
| C | | APRIORI CARD | SET02440 |
| C | 205 | IPRIOR = 1 | SET02450 |
| | | GO TO 10 | SET02460 |
| | 207 | J = NUMBER(CARD,COL,NUMVEC,0) | SET02470 |
| | | KREST = NUMVEC(1) | SET02480 |
| | | GO TO 10 | SET02490 |
| | 208 | J=NUMBER(CARD,COL,NUMVEC,0) | SET02500 |
| | | NCPASS=NUMVEC(1) | SET02510 |
| | | GO TO 10 | SET02520 |
| C* | | COMMENT CARD | SET02530 |
| C* | 210 | READ(30,6000)COMENT | SET02540 |
| | | REWIND RRUNIT | SET02550 |
| | | GO TO 10 | SET02560 |
| C* | | STAT FILE NO. | SET02570 |
| C* | | | SET02580 |
| C* | 215 | M = NXTCHR(CARD,COL) | SET02590 |
| | | IF(M.EQ.BLANK) GO TO 10 | SET02600 |
| | | IF(M.EQ.URCD) GO TO 1702 | SET02610 |
| | | IF(M.EQ.FRCD) GO TO 1703 | SET02620 |
| | 1723 | WRITE(6,755) | SET02630 |
| | 755 | FORMAT(' ERROR ON STAT FILE CARD *') | SET02640 |
| | | GO TO 10 | SET02650 |
| | 1702 | J=FIN12(CARD,COL,EQUVEC) | SET02660 |
| | | IF(J.EQ.-1) GO TO 1723 | SET02670 |
| | | M=NUMBER(CARD,COL,SAVTAP,ZERO) | SET02680 |
| | | COL=COL-1 | SET02690 |
| | | GO TO 215 | SET02700 |
| | 1703 | J=FIN12(CARD,COL,EQUVEC) | SET02710 |
| | | IF(J.EQ.-1) GO TO 1723 | SET02720 |
| | | FILNO = NUMBER(CARD,COL,STAFIL,FILNO) | SET02730 |
| | | STAFIL = STAFIL - 1 | SET02740 |
| | | COL=COL-1 | SET02750 |
| | | GO TO 215 | SET02760 |
| C* | | *END* - END OF THIS SET OF CONTROL CARDS | SET02770 |
| C* | | - GET STATS AND FETVEC INTO CORE | SET02780 |
| C* | 220 | CONTINUE | SET02790 |
| C* | | IF R-MATRIX IS INPUT, OBTAIN DIMENSIONING INFORMATION AND | SET02800 |
| C* | | FETVC2 FROM RMFILE. | SET02810 |
| C* | 225 | IF(RMSWT.EQ.0)GO TO 230 | SET02820 |
| | | CALL RMFIL (DUMMY,NOFET4,NOFET2,FETVC2,3) | SET02830 |
| C* | | READ AND REDUCE STATS | SET02840 |
| C* | 230 | CALL REDSAV(ARRAY,TOP,RMSWT) | SET02850 |
| C | | CODE ADDED TO CHECK FOR EXIT FOR ONE CLASS INPUT | SET02860 |
| C | | ON PROCEDURES 1,2,3,OR 6 | SET02870 |
| C | | NOFET4=NOFET2 | SET02880 |
| | | DO 240 I=1,NOFET2 | SET02890 |
| | | FETVC4(I)=FETVC2(I) | SET02900 |
| 240 | | CONTINUE | SET02910 |
| C | | | SET02920 |
| | | | SET02930 |
| | | | SET02940 |
| | | | SET02950 |
| | | | SET02960 |
| | | | SET02970 |
| | | | SET02980 |
| | | | SET02990 |
| | | | SET03000 |
| | | | SET03010 |
| | | | SET03020 |
| | | | SET03030 |
| | | | SET03040 |

FILE SETUP4

```

SURDS2=CLSID2
NA=SUBDS2 * NOSUR2
IWRDS = (VARSZ2*NOFET2)*NOSUB2
00 295 I=1,IWRDS
295 ARRAY(NA+I-1) = ARRAY(COVAR2+I-1)
COVAR2=NA
AVAR2=COVAR2 * NOSUB2*VARSZ2
C*
C* FROM HTER ON THROUGH SELECT SUBCLASSES ARE REFERRED TO AS CLASSES
C*
NOCLS2=NOSUB2
C*
C* COMPUTE BASES FOR OTHER ARRAYS.
C*
DIVSZ=NOCLS2*(NOCLS2-1)/2
WGHS14=AVAR2 * NOFET2*NOCLS2
DTAB4 = WGHS14 * DIVSZ
CORRAS = DTAB4 * DIVSZ*2
IC=CORRAS
IF(CORRAS.LT.TOP)GO TO 300
WRITE(6,9100)IC
CALL CMERR
C*
C* SET UP ARRAY OF INTERCLASS WEIGHTS IF INPUT - IF DEFAULT IS TAKEN
C* WEIGHTS ARE COMPUTED IN PRELIM.
C*
300 IF(SETWGT.NE.2.AND.WTKEY.NE.1) GO TO 310
C* PASS KEYS TO SUBROUTINE IN ALREADY EXISTING STORAGE
ARRAY(WGHS14) = SETWGT
NT = WGHS14 * 1
ARRAY(NT) = WTKEY
CALL WGTCHK(ARRAY(WGHS14),ARRAY(CLSID2),SUBRAY,WGMBUF,WPTR,
SUBRAY(W1),NOCLS2)
SETWGT = 2
310 CONTINUE
RETURN
C*
C* SEND* CARD
C*
350 STOPFG=1
RETURN
1000 FORMAT(' $SELECT')
2000 FORMAT(A4,5X,62A1)
3000 FORMAT(5X,A4,6X,62A1)
4000 FORMAT(' TOO MANY EVALUATE REQUESTS--REMAINDER IGNORED')
5000 FORMAT(' GROUP CARD IN ERROR - IGNORED')
6000 FORMAT(10X,15A4)
7000 FORMAT(' PROGRAM CANNOT PROCESS LESS THAN 2 CHANNELS')
8000 FORMAT(' INVALID CONTROL CARD - IGNORED')
9000 FORMAT(' PROGRAM CANNOT PROCESS LESS THAN 2 CLASSES')
9100 FORMAT(' CORE NEEDED IN ARRAY FOR THIS PROBLEM IS',I6,' WORDS')
9200 FORMAT(' ERROR IN ATTEMPT TO READ STATISTICS FILE-EXECUTION TERMINATED FROM SETUP4')
9300 FORMAT(' YOU HAVE SELECTED THE FOLLOWING OPTIONS:')
9310 FORMAT(5X,'PROCEDURE',T35,3A4/ 5X,'CRITERIA',T35,3A4)
9320 FORMAT(5X,'SELECT THE BEST SET(S) OF',T35,10(I2,''))
9330 FORMAT(5X,'FROM CHANNELS',T35,30(I2,''))
9340 FORMAT(5X,'USE INPUT WEIGHTS')
9350 FORMAT(5X,'USE DEFAULT WEIGHTS')
9360 FORMAT(5X,'INCLUDE IN THE BEST SET, CHANNELS',30(I2,''))
9370 FORMAT(5X,'USE AUTOMATIC INTERCLASS SUBCLASS WEIGHTS')
9380 FORMAT(5X,'USE APRIORI WEIGHTING TO MODIFY INTERSUBCLASS WTS')
9390 FORMAT(5X,'NUMBER CHANNELS PER PASS IS',T35,I5)
C* INITIALIZE ALL SUBCLASS WEIGHT PAIRS TO 0.0 IN WORKING ARRAY
9500 CONTINUE
DO 2100 IK=1,NOSUR2
DO 2100 JK=1,NOSUB2
IIDUM=(W1+IK-1+(JK-1)*NOSUB2)
IJDUM=(W1+JK-1+(IK-1)*NOSUR2)
SUBRAY(IIDUM)=0.0
SUBRAY(IJDUM)=0.0
2100 SUBRAY(IIDUM)=0.0
C*
C* REPLACE INTERCLASS SURCLASS PAIRS WITH WEIGHT = 1.0
C*
END1 = 0
NK = NOCLS2 - 1
DO 2200 KI=1,NK

```

SET03810
 SET03820
 SET03830
 SET03840
 SET03850
 SET03860
 SET03870
 SET03880
 SET03890
 SET03900
 SET03910
 SET03920
 SET03930
 SET03940
 SET03950
 SET03960
 SET03970
 SET03980
 SET03990
 SET04000
 SET04010
 SET04020
 SET04030
 SET04040
 SET04050
 SET04060
 SET04070
 SET04080
 SET04090
 SET04100
 SET04110
 SET04120
 SET04130
 SET04140
 SET04150
 SET04160
 SET04170
 SET04180
 SET04190
 SET04200
 SET04210
 SET04220
 SET04230
 SET04240
 SET04250
 SET04260
 SET04270
 SET04280
 SET04290
 SET04300
 SET04310
 SET04320
 SET04330
 SET04340
 SET04350
 SET04360
 SET04370
 SET04380
 SET04390
 SET04400
 SET04410
 SET04420
 SET04430
 SET04440
 SET04450
 SET04460
 SET04470
 SET04480
 SET04490
 SET04500
 SET04510
 SET04520
 SET04530
 SET04540
 SET04550
 SET04560

FILE SETUP4

```

JJ1=ARRAY(SURN02-1+KI)
START1 = END1 + 1
END1 = START1 + JJ1 - 1
DO 2300 I=START1,END1
  FND2 = END1
  D = KI + 1
DO 400 K2 = 0,NOCLS2
  JJ2=ARRAY(SURN02-1+K2)
  START2 = END2 + 1
  FND2 = START2 + JJ2 - 1
DO 500 N = START2,END2
  KDUM=(W1-1+I+(N-1)*NOSUR2)
  KKDUM=(W1-1+N+(I-1)*NOSUR2)
  SURRAY(KDUM)=1.0
  SURRAY(KKDUM)=1.0
500 CONTINUE
400 CONTINUE
2300 CONTINUE
2200 CONTINUE
GO TO 287
END

```

```

SET04570
SET04580
SET04590
SET04600
SET04610
SET04620
SET04630
SET04640
SET04650
SET04660
SET04670
SET04680
SET04690
SET04700
SET04710
SET04720
SET04730
SET04740
SET04750
SET04760
SET04770

```

FILE: TRACE

```
C* FUNCTION TRACE(A,B,N)
C* DOUBLE PRECISION TRACE
C* FUNCTION ROUTINE TO COMPUTE THE TRACE OF THE PRODUCT OF TWO
C* SYMMETRIC MATRICES, STORED IN SYMMETRIC NOTATION. THE DIMENSIONS
C* OF A AND B ARE  $N*(N+1)/2$ 
C*
DOUBLE PRECISION A,B,SUM,SUM1
DIMENSION A(1),B(1)
K=0
SUM1=0.0
DO 20 I=1,N
  M=I-1
  SUM=0.0
  IF (M.EQ.0) GO TO 15
  SUM=0.0
  DO 10 J=1,M
    K=K+1
10 SUM = SUM + A(K)*B(K)
15 K=K+1
    SUM1 = SUM1 + A(K)*B(K) + SUM*2.
20 CONTINUE
TRACE = SUM1
RETURN
END
```

FILE: TRNDIV

```

C* SURROUTINE TRNDIV(SPMR,COVMTX,AVEMTX,COVMT2,AVEMT2,
C*   WEIGHT,DIVTAB,
C*   WRKRY,IWRKSZ,IPART,PARTLS,BMAT,IFULL)
C*
C* SURROUTINE TO COMPUTE THE AVERAGE WEIGHTED TRANSFORMED
C* DIVERGENCE, AND PARTIALS WITH RESPECT TO B.
C*
C* IF IFULL=1 COMPUTE TRANSFORMED DIVERGENCE FOR ALL 'NOFET' CHANNELS
C* PARTIALS CANNOT BE COMPUTED WHEN IFULL=1.
C*
C* INCLUDE COMRK7.LIST
C* DOUBLE PRECISION SPMR
C* INCLUDE COMRK1.LIST
C* COMMON/INFORM/NOCLS2,NOSUR2,NOFET2,VARSZ2,TOTVT2,NOFLD2,
C*   AVAR2,COVAR2,CLSID2,SURNO2,SURDS2,FLDSV2,VERTX2,
C*   FETVC2(30),SURVC2(75),SURPTR(75),CLSV2(60),
C*   KEPPTS(60),NOGRP,GRPNAM(60),GRPDEX(61),
C*   GRPCHK(61),GROUPS(124)
C* COMMON/FSL/CFAC,TOTMSR,SEPMSR,PRCKEY,CRIKEY,INCFET,
C*   INCVEC(30),ICOUNT,SETHGT,EVALRF(100),FETVC4(30)
C*   ,NOFET4,VARSZ4,CORHAS,UTAB4,WGHS14,RESTVC(10),DIVSZ
C*   ,STATKY,ADRESO,ADRESP,ADRESF,ADRSH1,ADRSH2
C*   ,INTEGER ADRESO,ADRESP,ADRESF,ADRSH1,ADRSH2,STATKY
C*   ,DOUBLE PRECISION CFAC,TOTMSR,SEPMSR
C*
C* SEND
C*   INTEGER VARSZ4
C*   INTEGER VARSZ2,DIVSZ
C*   DOUBLE PRECISION BMAT,PARTLS
C*   DOUBLE PRECISION DIVTAB,DET,DET2,CON,TRACE
C*   DOUBLE PRECISION COVMT2,AVEMT2,WRKRY(1),T(30)
C*   DIMENSION COVMTX(VARSZ2,NOCLS2), COVMT2(VARSZ4,NOCLS2),
C*   ,AVEMTX(NOFET2,NOCLS2), AVEMT2(NOFET4,NOCLS2),
C*   ,WEIGHT(DIVSZ), DIVTAB(DIVSZ), PARTLS(1), BMAT(1)
C*
C*   IVSZ=VARSZ4
C*   NF=NOFET4
C*   IF(IFULL.EQ.1) IVSZ=VARSZ2
C*   IF(IFULL.EQ.1) NF=NOFET2
C*   ICOV1=1
C*   ICOV2=ICOV1+IVSZ
C*   IS2=ICOV2+IVSZ
C*   IW1=IS2 + IVSZ
C*   IS1=IW1+IVSZ
C*   ITEST=IS1
C*   IF(IPART.LT.0) GO TO 3
C*   ZERO PARTIALS
C*   IQ=NOFET2*NOFET4
C*   DO 2 IK=1,IQ
C*     PARTLS(IK)=0.0
C*     IW2=IS1+VARSZ2
C*     IW3=IW2+IQ
C*     IW4=IW3+IQ
C*     ITEST=IW4+IQ
C*   3 CONTINUE
C*   IF(IWRKSZ/2.GE.ITEST) GO TO 1
C*   WRITE(6,600) IWRKSZ
C*   CALL CMERR
C*   1 CONTINUE
C*   SPMR=0.0
C*   MN=0
C*   NC=NOCLS2-1
C*   DO 100 I=1,NC
C*     NS=I+1
C*     FIND INVERSE COVAR FOR CLASS I
C*     DO 5 II=1,IVSZ
C*       IF(IFULL.EQ.1) WRKRY(II)=COVMTX(II,I)
C*       IF(IFULL.NE.1) WRKRY(II)=COVMT2(II,I)
C*     5 CONTINUE
C*     CALL COLINV(WRKRY(ICOV1),NF,IERR,3,DET)
C*     IF(IERR.EQ.0) GO TO 6
C*     WRITE(6,500) I
C*     GO TO 100
C*   6 DO 7 J=NS,NOCLS2
C*     DO 7 II=1,NF
C*       IF(IFULL.EQ.1) T(II)=AVEMTX(II,I)-AVEMTX(II,J)
C*       IF(IFULL.NE.1) T(II)=AVEMT2(II,I)-AVEMT2(II,J)
C*     7 CONTINUE
C*     MN=MN+1
C*     K=0

```

TRN00010
 TRN00020
 TRN00030
 TRN00040
 TRN00050
 TRN00060
 TRN00070
 TRN00080
 TRN00090
 TRN00100
 TRN00110
 TRN00120
 TRN00130
 TRN00140
 TRN00150
 TRN00160
 TRN00170
 TRN00180
 TRN00190
 COM00010
 COM00020
 COM00030
 COM00040
 COM00050
 COM00060
 TRN00270
 TRN00280
 TRN00290
 TRN00300
 TRN00310
 TRN00320
 TRN00330
 TRN00340
 TRN00350
 TRN00360
 TRN00370
 TRN00380
 TRN00390
 TRN00400
 TRN00410
 TRN00420
 TRN00430
 TRN00440
 TRN00450
 TRN00460
 TRN00470
 TRN00480
 TRN00490
 TRN00500
 TRN00510
 TRN00520
 TRN00530
 TRN00540
 TRN00550
 TRN00560
 TRN00570
 TRN00580
 TRN00590
 TRN00600
 TRN00610
 TRN00620
 TRN00630
 TRN00640
 TRN00650
 TRN00660
 TRN00670
 TRN00680
 TRN00690
 TRN00700
 TRN00710
 TRN00720
 TRN00730
 TRN00740
 TRN00750
 TRN00760
 TRN00770
 TRN00780
 TRN00790
 TRN00800

FILE: TRNDIV

```

DO 12 II=1,NF
DO 12 IJ=1,II
K=K+1
IF (IFULL.NE.1) GO TO 10
WRKRY (IS2+K-1)=COVMTX(K,I)+COVMTX(K,J)+T(II)*T(IJ)
GO TO 12
10 WRKRY (IS2+K-1)=COVMT2(K,I)+COVMT2(K,J)+T(II)*T(IJ)
12 CONTINUE
C* IF PARTIALS ARE TO BE CALCULATED COMPUTE FULL 'S' MATRIX FOR
C* CLASSES I AND J
IF (IPART.LT.0) GO TO 25
DO 15 II=1,NOFET2
15 T(II)=AVENTX(II,I)-AVENTX(II,J)
K=0
DO 20 II=1,NOFET2
DO 20 IJ=1,II
K=K+1
20 WRKRY (IS1+K-1)=COVMTX(K,I)+COVMTX(K,J)+T(II)*T(IJ)
C* FIND INVERSE FOR CLASS J
DO 30 II=1,IVSZ
IF (IFULL.EQ.1) WRKRY (ICOV2+II-1)=COVMTX(II,J)
IF (IFULL.NE.1) WRKRY (ICOV2+II-1)=COVMT2(II,J)
30 CONTINUE
CALL COLINV(WRKRY(ICOV2),NF,IERR,3,DET2)
IF (IERR.EQ.0) GO TO 35
WRITE(6,500)J
GO TO 90
C* SUM INVERSES AND COMPUTE TRACE OF SUM * S2
35 DO 40 II=1,IVSZ
40 WRKRY (IW1+II-1)=WRKRY(ICOV1+II-1)+WRKRY(ICOV2+II-1)
DIVTAB(MN)=TRACE(WRKRY(IW1),WRKRY(IS2),NF)/2.-2.*NF
DIVTAB(MN)=DEXP(-DIVTAB(MN)/16.)
SPMSR=SPMSR+DIVTAB(MN)*WEIGHT(MN)
IF (IPART.LT.0) GO TO 90
C* COMPUTE PARTIALS
CALL MT1(RMAT,COVMTX(1,I),WRKRY(IW2),NOFET4,NOFET2)
CALL MT3(WRKRY(ICOV1),WRKRY(IW2),WRKRY(IW3),NOFET4,NOFET4,NOFET2,
1,0)
CALL MT3(WRKRY(IS2),WRKRY(IW3),WRKRY(IW2),NOFET4,NOFET4,NOFET2,
1,0)
CALL MT3(RMAT,WRKRY(IS1),WRKRY(IW4),NOFET4,NOFET2,NOFET2,0,1)
DO 42 IK=1,IQ
L=IK-1
42 WRKRY (IW4+L)=WRKRY (IW4+L)-WRKRY (IW2+L)
CALL MT3(WRKRY(ICOV1),WRKRY(IW4),WRKRY(IW2),NOFET4,NOFET4,
NOFET2,1,0)
CALL MT1(RMAT,COVMTX(1,J),WRKRY(IW3),NOFET4,NOFET2)
CALL MT3(WRKRY(ICOV2),WRKRY(IW3),WRKRY(IW4),NOFET4,NOFET4,
NOFET2,1,0)
CALL MT3(WRKRY(IS2),WRKRY(IW4),WRKRY(IW3),NOFET4,NOFET4,
NOFET2,1,0)
CALL MT3(RMAT,WRKRY(IS1),WRKRY(IW4),NOFET4,NOFET2,NOFET2,0,1)
DO 43 IK=1,IQ
L=IK-1
43 WRKRY (IW4+L)=WRKRY (IW4+L)-WRKRY (IW3+L)
CALL MT3(WRKRY(ICOV2),WRKRY(IW4),WRKRY(IW3),NOFET4,NOFET4,
NOFET2,1,0)
DO 44 IK=1,IQ
L=IK-1
44 WRKRY (IW2+L)=WRKRY (IW2+L)+WRKRY (IW3+L)
CON=WEIGHT(MN)*DIVTAB(MN)/(16.*NOCLS2)
DO 50 IK=1,IQ
PARTLS(IK)=PARTLS(IK)-CON*WRKRY(IW2+IK-1)
50 CONTINUE
90 CONTINUE
100 CONTINUE
SPMSR=SPMSR/NOCLS2
RETURN
500 FORMAT(' REDUCED COVARIANCE MATRIX FOR CLASS',I3,' IS NOT POSITIVE
DEFINITE')
600 FORMAT(' NOT ENOUGH WORK AREA IN TRNDIV -- IWRKSZ=',I5)
END

```

TRN00810
 TRN00820
 TRN00830
 TRN00840
 TRN00850
 TRN00860
 TRN00870
 TRN00880
 TRN00890
 TRN00900
 TRN00910
 TRN00920
 TRN00930
 TRN00940
 TRN00950
 TRN00960
 TRN00970
 TRN00980
 TRN00990
 TRN01000
 TRN01010
 TRN01020
 TRN01030
 TRN01040
 TRN01050
 TRN01060
 TRN01070
 TRN01080
 TRN01090
 TRN01100
 TRN01110
 TRN01120
 TRN01130
 TRN01140
 TRN01150
 TRN01160
 TRN01170
 TRN01180
 TRN01190
 TRN01200
 TRN01210
 TRN01220
 TRN01230
 TRN01240
 TRN01250
 TRN01260
 TRN01270
 TRN01280
 TRN01290
 TRN01300
 TRN01310
 TRN01320
 TRN01330
 TRN01340
 TRN01350
 TRN01360
 TRN01370
 TRN01380
 TRN01390
 TRN01400
 TRN01410
 TRN01420
 TRN01430
 TRN01440
 TRN01450
 TRN01460
 TRN01470
 TRN01480
 TRN01490
 TRN01500
 TRN01510
 TRN01520
 TRN01530
 TRN01540
 TRN01550

FILE: TRNSFR

| | | |
|-------|--|----------|
| | SUBROUTINE TRNSFR(A,A2,W,BMAT) | TRN00010 |
| C | INCLUDE COMRK7.LIST | TRN00020 |
| C | INCLUDE COMRK1.LIST | TRN00030 |
| | COMMON/INFORM/NOCLS2,NOSUR2,NOFET2,VAR5Z2,TOTVT2,NOFLD2, | TRN00040 |
| | AVAR2,COVAR2,CLSID2,SURN02,SUR052,FLDSV2,VERTX2, | TRN00050 |
| | FETVC2(30),SUHVC2(75),SURPTR(75),CLSV2(60), | TRN00060 |
| | KEPPTS(40),NOGRP,GRPNAM(60),GRFDEX(41), | TRN00070 |
| | GRPCHK(41),GROUPS(124) | TRN00080 |
| | COMMON/FSL/CFAC,TOTMSR,SEPMSR,PHCKEY,CHKEY,INCFET, | COM00010 |
| | INCVEC(30),ICOUNT,SETWGT,EVALHF(100),FETVC4(30) | COM00020 |
| | ,NOFET4,VAR5Z4,CORHAS,DTAH4,WGHS14,RESTVC(10),DIVSIZ | COM00030 |
| | ,STATKY,ADRESO,ADRESP,ADRESF,ADRS1,ADRS2 | COM00040 |
| | INTEGER ADRESO,ADRESP,ADRESF,ADRS1,ADRS2,STATKY | COM00050 |
| | DOUBLE PRECISION CFAC,TOTMSR,SEPMSR | COM00060 |
| CSEND | INTEGER VAR5Z4,VAR5Z2 | TRN00160 |
| | DOUBLE PRECISION SUM | TRN00170 |
| | DOUBLE PRECISION RMAT(NOFET4,NOFET2) | TRN00180 |
| | DOUBLE PRECISION A2(VAR5Z4,NOCLS2),W (NOFET4,NOFET2) | TRN00190 |
| | DIMENSION A(VAR5Z2,NOCLS2) | TRN00200 |
| C* | | TRN00210 |
| C* | MULTIPLY BMAT * A * BMAT(TRANPOSE) AND STORE IN A2 | TRN00220 |
| C* | | TRN00230 |
| | DO 140 JJ=1,NOCLS2 | TRN00240 |
| | DO 150 I=1,NOFET4 | TRN00250 |
| | DO 150 J=1,NOFET2 | TRN00260 |
| | SUM=0.0 | TRN00270 |
| | DO 140 K=1,NOFET2 | TRN00280 |
| | IF(K.GE.J) IP=K*(K-1)/2 + J | TRN00290 |
| | IF(K.LT.J) IP=J*(J-1)/2 + K | TRN00300 |
| 140 | SUM=SUM + BMAT(I,K) * A(IP,JJ) | TRN00310 |
| | W(I,J)=SUM | TRN00320 |
| 150 | CONTINUE | TRN00330 |
| C* | | TRN00340 |
| | DO 160 I=1,NOFET4 | TRN00350 |
| | DO 170 J=1,NOFET2 | TRN00360 |
| | SUM=0.0 | TRN00370 |
| | DO 140 K=1,NOFET2 | TRN00380 |
| 160 | SUM=SUM + W(J,K) * BMAT(I,K) | TRN00390 |
| | IF(J.GE.I) IP=J*(J-1)/2 + I | TRN00400 |
| | IF(J.LT.I) IP=I*(I-1)/2 + J | TRN00410 |
| | A2(IP,JJ)=SUM | TRN00420 |
| 170 | CONTINUE | TRN00430 |
| 180 | CONTINUE | TRN00440 |
| 190 | CONTINUE | TRN00450 |
| | RETURN | TRN00460 |
| | END | TRN00470 |
| | | TRN00480 |

FILE: USERIN

```

C* SURROUTINE USERIN(COVMTX,AVEMTX,DIVTAB,WEIGHT,COVMT2,AVEMT2,S,S2,  USE00010
C*      BMAT,WRKRY,IWRKSZ)  USE00020
C*  USE00030
C*  SURROUTINE USERIN COORDINATES THE NECESSARY ROUTINES TO COMPUTE  USE00040
C*  THE REQUESTED SEPARABILITY MEASURE FOR THE INPUT B-MATRIX.  USE00050
C*  USE00060
C*  INCLUDE COMRK1.LIST  USE00070
C*  INCLUDE COMRK7.LIST  USE00080
C*  COMMON/INFORM/NOCLS2,NOSUR2,NOFET2,VARSZ2,TOTVT2,NOFLD2,  COM00010
C*      AVAR2,COVAR2,CLSID2,SURN02,SURDS2,FLDSV2,VERTX2,  COM00020
C*      FETVC2(30),SUHVC2(75),SUHPT2(75),CLSV2(60),  COM00030
C*      KEPPTS(60),NOGRP,GRPNAM(60),GRPDEX(61),  COM00040
C*      GRPCHK(61),GROUPS(124)  COM00050
C*  COMMON/FSL/CFAC,TOTMSR,SEPMSR,PRCKEY,CRIKEY,INCFET,  COM00010
C*      INCVEC(30),ICOUNT,SETWGT,EVALBF(100),FETVC4(30)  COM00020
C*      .NOFET4,VARSZ4,COWH4,DIAR4,WGHS14,RESTVC(10),DIVSIZ  COM00030
C*      .STATKY,ADRES0,ADRESP,ADRESF,ADRSH1,ADRSH2  COM00040
C*  INTEGER ADRES0,ADRESP,ADRESF,ADRSH1,ADRSH2,STATKY  COM00050
C*  DOUBLE PRECISION CFAC,TOTMSR,SEPMSR  COM00060
C$END  USE00100
C*  INTEGER CRIKEY,VARSZ2,VARSZ4  USE00110
C*  DIMENSION COVMTX(VARSZ2,NOCLS2), AVEMTX(NOFET2,NOCLS2),  USE00120
C*      S(VARSZ2,NOCLS2),  USE00130
C*      WEIGHT(1),WRKRY(1)  USE00140
C*  DOUBLE PRECISION COVMT2(VARSZ4,1),AVEMT2(NOFET4,1),  USE00150
C*      S2(VARSZ4,1),BMAT(1)  USE00160
C*  DOUBLE PRECISION DIVTAB(1)  USE00170
C*  DIMENSION DUM(1)  USE00180
C*  USE00190
C*  GET B-MATRIX FROM FILE IN SINGLE PRECISION THEN STORE IN D.P.  USE00200
C*  USE00210
C*  CALL BMFIL(WRKRY,NOFET4,NOFET2,FETVC2,2)  USE00220
C*  IK=NOFET4*NOFET2  USE00230
C*  DO 10 I=1,IK  USE00240
C*      10 BMAT(I)=WRKRY(I)  USE00250
C*  USE00260
C*  GET TRANSFORMED STATISTICS  USE00270
C*  USE00280
C*  CALL GTSTAT(COVMTX,AVEMTX,S,COVMT2,AVEMT2,S2,DUM,BMAT,WRKRY,  USE00290
C*      IWRKSZ)  USE00300
C*  EVALUATE SEPARABILITY MEASURE  USE00310
C*  IPART=-1  USE00320
C*  CALL EVALSP(SEPMSR,COVMTX,AVEMTX,S,COVMT2,AVEMT2,S2,DIVTAB,  USE00330
C*      WEIGHT,IPART,DUM,BMAT,WRKRY,IWRKSZ)  USE00340
C*  IF(CRIKEY.NE.1)RETURN  USE00350
C*  USE00360
C*  EVALUATE INTERCLASS DIVERGENCES  USE00370
C*  USE00380
C*  CALL DIVRG1(COVMT2,VARSZ4,AVEMT2,DIVTAB,NOCLS2,NOFET4,  USE00390
C*      WRKRY,IWRKSZ)  USE00400
C*  RETURN  USE00410
C*  END  USE00420

```

FILE: WGTCHK

```

SUBROUTINE WGTCHK(WEIGHT,CLSNAM,NAMPR,WGHT,WPTR,WRKRY,NOCLS2)
IMPLICIT INTEGER(A-Z)
REAL WEIGHT(1),WGHT(1),WRKRY(NOCLS2,NOCLS2),WHT
DIMENSION CLSNAM(NOCLS2),NAMPR(2,WPTR)
REAL RWGR,RKEY
LOGICAL*1 LTEMP(4),LSTOR(4),NSTOR(4)
EQUIVALENCE (RWGT,SETWGT),(RKEY,WTKEY),(LSTORE,LSTOR(1)),(NSTORE,
* NSTOR(1)),(ITEMP,LTEMP(1))
DATA BLANK/' ',OTHERS/'OTHER',BLANKS/' '
C* PASS KEYS IN ALREADY EXISTING STORAGE
RWGT = WEIGHT(1)
RKEY = WEIGHT(2)
C* DELETE BLANKS FOR TESTING
DO 20 I=1,NOCLS2
LSTORE = CLSNAM(I)
NSTORE = BLANKS
IZ = 1
DO 10 IY=1,4
ITEMP=BLANKS
ITEMP(IY)=LSTORE(IY)
IF((ITEMP.EQ.BLANK).AND.(IZ.EQ.1))GO TO 10
NSTOR(IZ) = LSTORE(IY)
IZ = IZ + 1
10 CONTINUE
20 CLSNAM(I) = NSTORE
IF(WTKEY.EQ.1) GO TO 27
C* SET ALL CLASS PAIR WEIGHTS TO 1.0 IF USER HAS NOT INPUT WEIGHT FOR
C* 'OTHERS' OR 'CLSNAM'. SET TO 'OTHERS' VALUE IF INPUT.
WHT=1.0
DO 22 I=1,WPTR
IF(NAMPR(I,1).NE.OTHERS)GO TO 22
WHT=WGHT(1)
GO TO 24
22 CONTINUE
24 DO 25 IK=1,NOCLS2
DO 25 JK=1,NOCLS2
WRKRY(IK,JK)=WHT
25 WRKRY(JK,IK)=WHT
C* PLACEMENT OF INPUTED WEIGHT VALUES
27 IF(SETWGT.NE.2) GO TO 55
DO 50 I = 1,WPTR
DO 40 J=1,NOCLS2
IF (CLSNAM(J).NE. NAMPR(1,I))GO TO 40
C* FOUND MATCH ON FIRST NAME IN NAMPR - INDEX J
C* NOW SEE ABOUT SECOND NAME
IF(NAMPR(2,I).EQ. BLANK) GO TO 35
DO 30 K=1,NOCLS2
IF (CLSNAM(K).NE. NAMPR(2,I))GO TO 30
C* FOUND SECOND MATCH - INDEX K
WRKRY(J,K)=WGHT(1)
WRKRY(K,J)=WGHT(1)
GO TO 50
30 CONTINUE
WRITE(6,100) NAMPR(2,I)
GO TO 50
C* ALL PAIRS FOR CLASS J SET TO SAME WEIGHT
35 DO 36 IK=1,NOCLS2
WRKRY(IK,J)=WGHT(1)
36 WRKRY(J,IK)=WGHT(1)
GO TO 50
40 CONTINUE
IF(NAMPR(1,I).EQ.OTHERS)GO TO 50
46 WRITE(6,100) NAMPR(1,I)
50 CONTINUE
55 CONTINUE
C* REDUCE WRKRY MATRIX AND STORE IN WEIGHTS
K=0

```

WGT00010
WGT00020
WGT00030
WGT00040
WGT00050
WGT00060
WGT00070
WGT00080
WGT00090
WGT00100
WGT00110
WGT00120
WGT00130
WGT00140
WGT00150
WGT00160
WGT00170
WGT00180
WGT00190
WGT00200
WGT00210
WGT00220
WGT00230
WGT00240
WGT00250
WGT00260
WGT00270
WGT00280
WGT00290
WGT00300
WGT00310
WGT00320
WGT00330
WGT00340
WGT00350
WGT00360
WGT00370
WGT00380
WGT00390
WGT00400
WGT00410
WGT00420
WGT00430
WGT00440
WGT00450
WGT00460
WGT00470
WGT00480
WGT00490
WGT00500
WGT00510
WGT00520
WGT00530
WGT00540
WGT00550
WGT00560
WGT00570
WGT00580
WGT00590
WGT00600
WGT00610
WGT00620
WGT00630
WGT00640
WGT00650
WGT00660
WGT00670
WGT00680
WGT00690
WGT00700
WGT00710
WGT00720
WGT00730
WGT00740
WGT00750
WGT00760
WGT00770
WGT00780
WGT00790

FILE: WGTCHK

```
NC=NOCLS2-1
DO 60 I=1.NC
  IK=I+1
  DO 60 J=IK.NOCLS2
    K=K+1
    WEIGHT(K)=WPKRY(I,J)
  60 CONTINUE
  RETURN
100 FORMAT(1 SURCLASS 'A6' IS NOT AMONG INPUT SUBCLASSES - WEIGHT IN
  *PUT IGNORED*)
END
```

WGT00800
WGT00810
WGT00820
WGT00830
WGT00840
WGT00850
WGT00860
WGT00870
WGT00880
WGT00890
WGT00900

FILE: WGTSCN

```

SUBROUTINE WGTSCN(CARD,COL,NAMPR,WGHT,WSIZ,NCNT)
IMPLICIT INTEGER(A-Z)
-----
CALL... J=WGTSCN(CARD,COL,NAMPR,WGHT,WSIZ)
-----
ARGS... CARD - ARRAY OF CHARACTERS TO BE SCANNED.
          COL - COLUMN IN CARD TO BEGIN SCAN, ON OUTPUT
          COL IS LAST COLUMN OF CARD SCANNED.
          NAMPR- ARRAY CONTAINING, ON OUTPUT, THE PAIRS OF CLASS
          NAMES SCANNED FROM CARD.
          WGHT - ARRAY CONTAINING WEIGHT FOR CORRESPONDING
          CLASS PAIR
          WSIZ - SIZE OF WGHT BUFFER
          NCNT - RUNNING COUNT OF NAME PAIRS SCANNED
          PURPOSE- SCANS THE WEIGHTS CONTROL CARD, SAVING THE CLASS NAME
          PAIRS AND ASSOCIATED WEIGHT FOR LATER VERIFICATION IN
          SUBROUTINE WGTCHK. THE WEIGHT CARD MAY TAKE THE FOLLOWIN
          FORMS:
          WEIGHT CLASS 1=10.5, CLASS 2=12.0, OTHERS=20.0
          OR:
          WEIGHTS (CLASS 1, CLASS 2)=15.0, CLASS 3=1.0, OTHERS=5
-----
DIMENSION COMVEC(2),EQUVEC(2),RPNVEC(2)
DATA BLANKS/' ',LEFTPRN/'(',COMMA/',',EQUAL/'=',RIGHTPRN/')'/
DATA BLANKS/' '/
DATA COMVEC/1,' ',EQUVEC/1,' ',RPNVEC/1,' '/
REAL WGHT
DIMENSION CARD(1),NAMPR(2,1),WGHT(1)
LOGICAL*1 LSTOR(4),LCARD(4)
EQUIVALENCE (MCARD,LCARD(1)),(STOR,LSTOR(1))
C*
C* 1 J=NXTCHR(CARD,COL)
C* IF(J.EQ.BLANK)GO TO 60
C* IF(J.NE.LEFTPRN)GO TO 2
C* COL=COL+1
C* 2 CONTINUE
C* BLANK OUT NAMPR FOR THIS PAIR
C* NCNT=NCNT+1
C* DO 3 I=1,2
C* 3 NAMPR(I,NCNT)=BLANKS
C* IR=WSIZ-NCNT
C* IF(IR.GT.0)GO TO 4
C* WRITE(4,200)WSIZ
C* GO TO 60
C* 4 CONTINUE
C* STOR=BLANKS
C* WCNT = 0
C* 6 WCNT = WCNT + 1
C* IF(CARD(COL).EQ.BLANK)GO TO 10
C* IF(CARD(COL).EQ.EQUAL)GO TO 45
C* IF(CARD(COL).EQ.COMMA)GO TO 20
C* MCARD = CARD(COL)
C* LSTOR(WCNT)=LCARD(1)
C* NAMPR(1,NCNT)=STOR
C* 10 COL=COL+1
C* IF(WCNT.EQ.4)GO TO 15
C* GO TO 6
C*
C* ONLY FOUR CHARACTERS PER NAME ALLOWED - IGNORE REMAINDER
C* FIND = 0 - ERROR OTHERWISE
C*
C* 15 CONTINUE
C* J=IND12(CARD,COL,COMVEC)
C* IF(J.EQ.-1)GO TO 40
C* GO TO 19
C*
C* COMMA FOUND - ANOTHER NAME SHOULD FOLLOW
C*
C* 19 WCNT = 0
C* 20 COL=COL+1
C* 21 WCNT = WCNT + 1
C* STOR = BLANKS
C* IF(CARD(COL).EQ.RIGHTPRN)GO TO 40
C* IF(CARD(COL).EQ.BLANK)GO TO 25

```

WGT00010
 WGT00020
 WGT00030
 WGT00040
 WGT00050
 WGT00060
 WGT00070
 WGT00080
 WGT00090
 WGT00100
 WGT00110
 WGT00120
 WGT00130
 WGT00140
 WGT00150
 WGT00160
 WGT00170
 WGT00180
 WGT00190
 WGT00200
 WGT00210
 WGT00220
 WGT00230
 WGT00240
 WGT00250
 WGT00260
 WGT00270
 WGT00280
 WGT00290
 WGT00300
 WGT00310
 WGT00320
 WGT00330
 WGT00340
 WGT00350
 WGT00360
 WGT00370
 WGT00380
 WGT00390
 WGT00400
 WGT00410
 WGT00420
 WGT00430
 WGT00440
 WGT00450
 WGT00460
 WGT00470
 WGT00480
 WGT00490
 WGT00500
 WGT00510
 WGT00520
 WGT00530
 WGT00540
 WGT00550
 WGT00560
 WGT00570
 WGT00580
 WGT00590
 WGT00600
 WGT00610
 WGT00620
 WGT00630
 WGT00640
 WGT00650
 WGT00660
 WGT00670
 WGT00680
 WGT00690
 WGT00700
 WGT00710
 WGT00720
 WGT00730
 WGT00740
 WGT00750
 WGT00760
 WGT00770
 WGT00780
 WGT00790

FILE: WGTSCN

| | |
|--|----------|
| MCARD = CARD(COL) | WGT00800 |
| LSTOR(WCNT) = LCARD(1) | WGT00810 |
| NAMPR(2,NCNT)=STOR | WGT00820 |
| 25 COL=COL+1 | WGT00830 |
| IF(WCNT.EQ.4)GO TO 30 | WGT00840 |
| GO TO 21 | WGT00850 |
| 30 CONTINUE | WGT00860 |
| J=FINI2(CARD,COL,RPNVEC) | WGT00870 |
| IF(J.NE.-1)GO TO 40 | WGT00880 |
| WRITE(6,100) | WGT00890 |
| NCNT=NCNT-1 | WGT00900 |
| GO TO 60 | WGT00910 |
| C* | WGT00920 |
| C* | WGT00930 |
| C* | WGT00940 |
| 40 J=FINI2(CARD,COL,EQUVEC) | WGT00950 |
| IF(J.NE.-1)GO TO 45 | WGT00960 |
| WRITE(6,100) | WGT00970 |
| NCNT=NCNT-1 | WGT00980 |
| GO TO 40 | WGT00990 |
| 45 J=FLTHM(CARD,COL,WGHT(NCNT),IR) | WGT01000 |
| COL=COL-1 | WGT01010 |
| J=NXTCHW(CARD,COL) | WGT01020 |
| IF(J.EQ.COMMA)GO TO 1 | WGT01030 |
| IF(J.EQ.BLANK)GO TO 60 | WGT01040 |
| IF(J.NE.LFIPRN)GO TO 2 | WGT01050 |
| COL=COL+1 | WGT01060 |
| GO TO 2 | WGT01070 |
| 40 CONTINUE | WGT01080 |
| RETURN | WGT01090 |
| 100 FORMAT(' SYNTAX ERROR ON WEIGHT CARD-REMAINDER OF CARD IGNORED') | WGT01100 |
| 200 FORMAT(' WEIGHT BUFFER IS FILLED-ONLY',IS,' CLASS NAME PAIRS ALLOW | WGT01110 |
| 'ED') | WGT01120 |
| END | WGT01130 |

FILE: WHRPLC

```

SUBROUTINE WHRPLC(COVMTX,AVEMTX,DIVTAB,WEIGHT,COVMT2,
    AVEMT2,S,S2,WKRY,IWKRSZ)
C*
C* SUBROUTINE TO FIND THE BEST SET OF NOFET4 FEATURES
C* USING THE WITHOUT REPLACEMENT PROCEDURE.
C*
C* INTEGER FETVC2,FETVC4,TVEC,TRYVEC,KEEP
C* INCLUDE COMARK7.LIST
C* INCLUDE COMARK1.LIST
C* COMMON/INFORM/NOCLS2,NOSUR2,NOFET2,VARSZ2,TOTVT2,NOFLD2,
    AVAR2,CVAR2,CLS1D2,SURNO2,SUBDS2,FLDSV2,VERTX2,
    FETVC2(30),SURVC2(75),SURPTH(75),CLSV2(60),
    KEMPTS(60),NOGRP,GWPNAM(60),GRPDEX(61),
    GRPCHK(61),GROUPS(124)
C* COMMON/FSL/CFAC,TOTMSR,SEPMSR,PRCKEY,CRIKEY,INCFET,
    INCVEC(30),ICOUNT,SETWGT,EVALHF(100),FETVC4(30),
    NOFET4,VARSZ4,COPHAS,DTR4,WGMS14,HESTVC(10),DIVSIZ
    ,STATKY,ADRES0,ADRES1,ADRESF,ADRSN1,ADRSN2
C* INTEGER ADRES0,ADRES1,ADRESF,ADRSN1,ADRSN2,STATKY
C* DOUBLE PRECISION CFAC,TOTMSR,SEPMSR
CSEND
    DOUBLE PRECISION COVMT2(1),AVEMT2(1),S2(1)
    DOUBLE PRECISION DIVTAB(1),TMSR,DUM(1),DM
    INTEGER CRIKEY
    DIMENSION COVMTX(1),AVEMTX(1),WEIGHT(1),
    S(1),WKRY(1)
    IPART=-1
    DIMENSION TVEC(30),NREST(30),TRYVEC(30)
C*
C* SAVE THE VALUE OF NOFET4
    NFAVE=NOFET4
    IF(NRST.GT.0)GO TO 15
    IF(INCFET.LE.0)GO TO 15
    DO 10 I=1,INCFET
    DO 5 J=1,NOFET2
    IF(INCVEC(I).EQ.FETVC2(J))GO TO 6
5 CONTINUE
    WRITE(6,100)INCVEC(I)
    GO TO 10
10 NRST=NRST+1
    NREST(NRST)=J
10 CONTINUE
C*
C* SET UP VECTOR OF FEATURES TO TRY WITH NREST
C*
15 IF(NRST.GE.NFAVE)GO TO 50
    SEPMSR=1.E+35
    NTRY=0
    DO 25 I=1,NOFET2
    IF(NRST.EQ.0)GO TO 24
    DO 20 J=1,NRST
    IF(I.EQ.NREST(J))GO TO 25
20 CONTINUE
24 NTRY=NTRY+1
    TRYVEC(NTRY)=I
25 CONTINUE
C*
C* TRY EACH FEATURE IN TRYVEC WITH THE 'NREST' SO FAR AND KEEP
C* THE ONE WHICH GIVES MAXIMUM SEPARABILITY MEASURE.
C*
    NF=NRST+1
    NOFET4=NF
    DO 40 I=1,NTRY
    IF(NRST.EQ.0)GO TO 35
    DO 30 J=1,NRST
    TVEC(J)=NREST(J)
30 TVEC(NF)=TRYVEC(I)
35 CALL ORDER(TVEC,NF)
    CALL GSTAT(COVMTX,AVEMTX,S,COVMT2,AVEMT2,S2,TVEC,DM,WKRY,IWKRSZ)
    CALL EVALSP(TMSR,COVMTX,AVEMTX,S,COVMT2,AVEMT2,S2,DIVTAB,
    WEIGHT,IPART,DUM,DUM,WKRY,IWKRSZ)
    IF(SEPMSR.LT.TMSR)GO TO 40
    KEEP=TRYVEC(I)
    SEPMSR=TMSR
40 CONTINUE
    NRST=NRST+1
    NREST(NRST)=KEEP
    GO TO 15
50 NOFET4=NFAVE

```

WHR00010
WHR00020
WHR00030
WHR00040
WHR00050
WHR00060
WHR00070
WHR00080
WHR00090
WHR00100
WHR00110
WHR00120
WHR00130
WHR00140
COM00010
COM00020
COM00030
COM00040
COM00050
COM00060
WHR00220
WHR00230
WHR00240
WHR00250
WHR00260
WHR00270
WHR00280
WHR00290
WHR00300
WHR00310
WHR00320
WHR00330
WHR00340
WHR00350
WHR00360
WHR00370
WHR00380
WHR00390
WHR00400
WHR00410
WHR00420
WHR00430
WHR00440
WHR00450
WHR00460
WHR00470
WHR00480
WHR00490
WHR00500
WHR00510
WHR00520
WHR00530
WHR00540
WHR00550
WHR00560
WHR00570
WHR00580
WHR00590
WHR00600
WHR00610
WHR00620
WHR00630
WHR00640
WHR00650
WHR00660
WHR00670
WHR00680
WHR00690
WHR00700
WHR00710
WHR00720
WHR00730
WHR00740
WHR00750
WHR00760
WHR00770
WHR00780
WHR00790
WHR00800

FILE: WHRPLC

```
      DO 60 I=1,NOFET4
      K=NRST(I)
      TVEC(I)=K
C* 60 FETVC4(I)=FETVC2(K)
C*
C*   COMPUTE INTERCLASS MEASURES FOR FEATURES CHOSEN.
C*
      CALL ORDER(TVEC,NOFET4)
      CALL GTSTAT(COVMTX,AVEMTX,S,COVMT2,AVEMT2,S2,TVEC,DUM,WRKRY,
      * IWRKSZ)
      * CALL FVALSP(SFPMSP,COVMTX,AVEMTX,S,COVMT2,AVEMT2,S2,DIVTAB,
      * WEIGHT,IPART,DUM,DUM,WRKRY,IWRKSZ)
      * IF(CRIKEY.NF.1)RETURN
      * CALL DIVRG1(COVMT2,VAR24,AVEMT2,DIVTAB,NOCLS2,
      * NOFET4,WRKRY,IWRKSZ)
      * RETURN
100 FORMAT(' THE INCLUDE REQUEST FOR FEATURE',I4,
      * ' IS NOT A LEGITIMATE REQUEST--IGNORED')
      END
```

WHR00810
WHR00820
WHR00830
WHR00840
WHR00850
WHR00860
WHR00870
WHR00880
WHR00890
WHR00900
WHR00910
WHR00920
WHR00930
WHR00940
WHR00950
WHR00960
WHR00970
WHR00980
WHR00990

ORIGINAL PAGE IS
OF POOR QUALITY

10-66

136

11. CLASSIFY PROCESSOR

FILE: CLSFY

```

SURROUTINE CLSFY (ARRAY, TOP)
IMPLICIT INTEGER (A-H, O-Z)
DIMENSION ARRAY (3000)
-----
CALL..    CALL CLSFY (ARRAY, TOP)
ARGS..    ARRAY = SFE 'MONITOR'
          TOP = SFE 'MONITOR'
REQUIRES. COMMONS /INFORM/CLASS/GLOBAL/BMTRX/SCRACH/
          ROUTINES SETUP2 CLSFY1 CLSFY2
PURPOSE..  COORDINATES THE VARIOUS ROUTINES
          FOR 'CLASSIFICATION' STEP
RETURNS..  NONE
-----
INCLUDE COMBK1, LIST
-----
INCLUDE COMBK2, LIST
INCLUDE COMBK6, LIST
COMMON /INFORM/ NOCLS2, NOSUR2, NOFET2, VARSZ2, TOTVT2, NOFLD2,
*          AVAR2, COVAR2, CLSID2, SUBNO2, SUBDS2, FLDV2, VERTX2,
*          FETVC2 (30), SURVC2 (75), SUBPTR (75), CLSVC2 (60),
*          KEPPTS (60), NOGRP, GRPNAM (60), GRPDEX (61),
*          GRPCHK (61), GROUPS (124)
CSEND
COMMON /CLASS/ APRFLG, BMCOMB, BMFEAT, BMFLG, NOCAT, THIJ1, IDATA1,
*          NFILE, STATKY, CATNAM (60),
*          CLSSYM (60), CON (60), DET (60), FLDESC, FLDINF (6),
*          KCLSNA (60), NOCTCL (60), SURCAT (60)
*          , NOCHAN, CHNVEC (30)
CLASS COMMON IS USED BY THE CLASSIFY PROCESSOR. IT IS IN CORE
ONLY WHEN THIS PROCESSOR IS.
APRFLG - SET TO THE NUMBER OF APRIORI VALUES READ IN FROM
        INPUT CARDS
BMCOMB - NUMBER OF LINEAR COMBINATIONS IN B-MATRIX
BMFEAT - NUMBER OF CHANNELS USED IN COMPUTING THE B-MATRIX
BMFLG - INDICATES WHETHER A B-MATRIX HAS BEEN INPUT
NOCAT - NUMBER OF CATEGORIES
THIJ1 - BEGINNING ADDRESS FOR STORING THE CLASS-PAIR THRES-
        HOLD TABLE
IDATA1 - BEGINNING ADDRESS FOR STORING THE DATA
NFILE - NUMBER OF THE NEXT FILE TO BE WRITTEN ON MAPTAP
STATKY - FLAG CONTROLLING STATISTICS PRINT OUT
CATNAM - CONTAINS THE CATEGORY NAMES
CLSSYM - DEFAULT SYMBOLS USED IN PRINTING CLASSIFICATION MAP
CON - CONTAINS THE SURCLASS CONSTANTS
DET - CONTAINS THE SUBCLASS DETERMINANTS
FLDESC - FIELD NAME
FLDINF - CONTAINS THE RECTANGULAR COORDINATES SURROUNDING
        THE NON-RECTANGLE FIELD
KCLSNA - CONTAINS THE CLASS NAMES IN THE ORDER TAKEN OFF THE
        CATEGORY CONTROL CARD
NOCTCL - NUMBER OF CLASSES IN EACH CATEGORY
SURCAT - CONTAINS THE CATEGORY NUMBER TO WHICH THE CORRESPONDING
        SURCLASS BELONGS
COMMON /GLOBAL/ HEAD (63), MAPTAP, DATAPE, SAVTAP, BMFILE, BMKEY,
*          HISFIL, HISKEY, INFORM, ERIP, ERPKEY, MAPUNT, NOFILE,
*          DRUMADS, PAGES17, DATEFIL, STAFIL, ASAV, ASAVFL
*          , NHSTUN, NHSTFI, SCTRUN, MAPFIL
*          , DOTUNT, DOTFIL, NCHPAS, TRNSFL, BMTRFL, HISTFL, PCHUNT,
*          CRDUNT, PRUNT, HANDIO

```

FILE: CLSFY

```

COMMON /BMTRX/BMATRX(450)
-----
DIMENSION KATNO(60)
-----
REAL APRIOR(60)
-----
COMMON /SCRACH/ IDATA(12500)
-----
*** NOTE: THE IDATA ARRAY IS USED EXTENSIVELY AS A SCRATCH
AREA IN CLSFY1, FOR OUTPUT OF CLASS-PAIR THRESHOLDS IN
SURR. THRESH (FOR USE IN CONTEX), INPUT OF SCAN LINE TO BE
CLASSIFIED IN CLSFY2.

FLDFLG=0
CALL SETUP2(ARRAY.TOP,FLDFLG,APRIOR,BMATRX,KATNO)
IF (FLDFLG.EQ.1) GO TO 10

CALL CLSFY1(ARRAY(COVAR2),ARRAY(AVAR2),ARRAY(FLDSV2),
* ARRAY(CLSID2),APRIOR,BMATRX,ARRAY(VERTEX2),ARRAY(SUBDS2),
* ARRAY(SUBNO2),ARRAY(COVAR2),ARRAY(AVAR2),KATNO)

CALL CLSFY2(ARRAY(COVAR2),ARRAY(AVAR2),ARRAY(FLDSV2),
* ARRAY(CLSID2),ARRAY(SUBDS2),ARRAY(SUBNO2),KATNO,BMATRX)

END FILE MAPTAP

WRITE (6,20)
FORMAT(///// 1X, '*** SCLASSIFY - COMPLETED ***' /////)
RETURN
END

```

CLS00770
 CLS00780
 CLS00790
 CLS00800
 CLS00810
 CLS00820
 CLS00830
 CLS00840
 CLS00850
 CLS00860
 CLS00870
 CLS00880
 CLS00890
 CLS00900
 CLS00910
 CLS00920
 CLS00930
 CLS00940
 CLS00950
 CLS00960
 CLS00970
 CLS00980
 CLS00990
 CLS01000
 CLS01010
 CLS01020
 CLS01030
 CLS01040
 CLS01050
 CLS01060
 CLS01070
 CLS01080
 CLS01090
 CLS01100
 CLS01110
 CLS01120
 CLS01130
 CLS01140
 CLS01150
 CLS01160
 CLS01170
 CLS01180
 CLS01190

FILE CATGRY

```

SUBROUTINE CATGRY(NCHAN,NPTS,AVE,COR,IR,VR,BMATR,IData,NLINE,
  VERTCS,NV,PTSTHS)
-----
      NCHAN : NO. OF CHANNELS
      NPTS  : NO. OF PTS. IN RECTANGLE FIELD
      AVE   : MEANS ARRAY
      COR   : COVARIANCE ARRAY
      IR    : CLASSIFIED DATA
      VR    : CORRESPONDING PDFS OF IR ARRAY
      CON   : SUBCLASS CONSTANTS
      BMATR : B-TRANSFORMATION MATRIX, IF AVAILABLE
      IDATA : SCAN LINE OF DATA TO BE CLASSIFIED
      NLINE : LINE NUMBER CORRESPONDING TO DATA TAPE
      VERTCS : VERTICES OF FIELD TO BE CLASSIFIED
      NV    : NO. OF VERTICES

      PURPOSE : EACH PIXEL IS ASSIGNED TO A CATEGORY, THEN ASSIGNED
                  TO A SUBCLASS WITHIN THAT CHOSEN CATEGORY. ON THE
                  MAPTAP THE CHOSEN SUBCLASS NUMBER AND ITS CORRESPOND-
                  ING PDF IS OUTPUT.

      RETURNS : IR ARRAY RETURNS THE SUBCLASS NUMBER EACH PIXEL WAS
                  ASSIGNED TO
                  : VR ARRAY RETURNS THE CORRESPONDING PDF
-----
      IMPLICIT INTEGER (A-Z)
      LOGICAL BMFLAG,KD1
      DIMENSION IDATA(1),IR(1000),FL(22),SUBNUM(60),VERTCS(1)
      REAL VR(1000),AVE(1),COR(1),DATA(30),DM(30),
      * BMATR(BMCOMB,BMFEAT),S,P(60),TF,PK
      REAL FDATA,SUM
      REAL IFMAX(60),F,CON
      C INCLUDE COMBK1,LIST
      C INCLUDE COMBK2,LIST
      COMMON/INFORM/NOCLS2,NOSUB2,NOFET2,VARSZ2,TOTVT2,NOFLD2,
      * AVAR2,COVAR2,CLSID2,SUBNO2,SUBDS2,FLDSV2,VERTX2,
      * FETVC2(30),SUHVC2(75),SUHPT(75),CLSV2(60),
      * KEPPTS(60),NOGRP,GRPNAM(60),GRPDEX(61),
      * GRPCHK(61),GROUPS(124)
      COMMON /CLASS/ APRFLG,BMCOMB,BMFEAT,BMFLG,NOCAT,THIJ1,IData1,
      * NFILE,STATKY,CATNAM(60),
      * CLSSYM(60),CON(60),DET(60),FLDESC,FLDINF(6),
      * KCLSNA(60),NOCTCL(60),SUBCAT(60)
      * ,NOCHAN,CHNVEC(30)
      CSEND EQUIVALENCE (FLDINF(1),LINSTR), (FLDINF(2),LINEND),
      * (FLDINF(3),LININC), (FLDINF(4),SAMSTR),
      * (FLDINF(5),SAMEND), (FLDINF(6),SAMINC)

      C
      BMFLAG = RMFLG .GT. 0
      IF (BMFLAG) GO TO 10
      NF = NCHAN
      GO TO 20
      NF = BMCOMB
      KD1 = NF .EQ. 1

      C
      ZERO OUT IR AND VR

      DO 55 K=1,NPTS
      IR(K) = 0
      VR(K) = 0.0

      C
      CALL FDLINT(VERTCS,NV,FL,NLINE,IPTS,NI)
      DO 350 JJ=1,NI,2
      IB = (FL(JJ) - SAMSTR) / SAMINC + 1
      IF = (FL(JJ+1) - SAMSTR) / SAMINC + 1
      IF (MOD(SAMSTR,SAMINC) .NE. MOD(FL(JJ),SAMINC)) IB = IB + 1

```

FILE CATGRY

```

23 IF (IR .GT. IE) GO TO 350
DO 250 II=IH,IE
C
C
C   FLOAT DATA SAMPLE, AND APPLY THE B-MATRIX, IF AVAILABLE
C   IF (BMFLAG) GO TO 30
DO 25 I=1,NCHAN
IDUM = NPTS * (I - 1) + II
25 DATA(I) = IDATA(IDUM)
GO TO 45
C
30 DO 35 I=1,BMCOMB
SUM = 0.0
DO 40 K=1,NCHAN
INDUM = NPTS * (K - 1) + I
FDATA = IDATA(INDUM)
SUM = SUM + BMATR(I,K) * FDATA
40 DATA(I) = SUM
35 CONTINUE
45
C
60 DO 65 LL=1,NOCAT
65 P(LL) = 0.0
C
C
C   IRM = - NF
LC = 0
DO 130 KL=1,NOCAT
130 TFMAX(KL) = -1.0E35
C
C
C   DO 150 KK=1,NOSUR2
IRM = IRM + NF
KM = IRM + 1
S = DATA(1) - AVE(KM)
DM(1) = S
LC = LC + 1
TF = CON(KK) + (S*S) / COR(LC)
IF (KD1) GO TO 146
C
C
C   LOOP FOR COMPUTING THE KD-TH ELEMENT OF Y ( L**-1 * (X-M) ), WHICH
C   IS STORED IN S
C
DO 145 KD=2,NF
KM = IRM + KD
S = DATA(KD) - AVE(KM)
J1 = KD - 1
DO 140 LD=1,J1
LC = LC + 1
140 S = S - COR(LC) * DM(LD)
C
C
C   DM(KD) = S
C
C   LC = LC + 1
C   COMPUTE THE KD-TH TERM IN : 1/2 * Y * D**-1 * Y
C   = 1/2 * (X-M) * K**-1 * (X-M)
C
145 TF = TF + ( S*S ) / COR(LC)
C
146 TF = -.5 * TF
C
C   SUM THE EXP(P(KK))
C
C   IF PDF IS SMALLER THAN -88, DO NOT EXP.
IF ( TF .LE. -88 ) E = 0.0
IF ( TF .LE. -88 ) GO TO 148
E = EXP(TF)
148 CONTINUE
CTGORY = SUBCAT(KK)
P(CTGORY) = E + P(CTGORY)
C
C
C   FIND MAX. VALUE OF PDF OVER ALL SUBCLASSES WITHIN A CATEGORY
C   AND SAVE THE SUBCLASS NUMBER OF THE LARGEST PDF
C
IF (TF .LE. TFMAX(CTGORY)) GO TO 150
TFMAX(CTGORY) = TF
SUBNUM(CTGORY) = KK
150 CONTINUE

```

CAT00740
 CAT00750
 CAT00760
 CAT00770
 CAT00780
 CAT00790
 CAT00800
 CAT00810
 CAT00820
 CAT00830
 CAT00840
 CAT00850
 CAT00860
 CAT00870
 CAT00880
 CAT00890
 CAT00900
 CAT00910
 CAT00920
 CAT00930
 CAT00940
 CAT00950
 CAT00960
 CAT00970
 CAT00980
 CAT00990
 CAT01000
 CAT01010
 CAT01020
 CAT01030
 CAT01040
 CAT01050
 CAT01060
 CAT01070
 CAT01080
 CAT01090
 CAT01100
 CAT01110
 CAT01120
 CAT01130
 CAT01140
 CAT01150
 CAT01160
 CAT01170
 CAT01180
 CAT01190
 CAT01200
 CAT01210
 CAT01220
 CAT01230
 CAT01240
 CAT01250
 CAT01260
 CAT01270
 CAT01280
 CAT01290
 CAT01300
 CAT01310
 CAT01320
 CAT01330
 CAT01340
 CAT01350
 CAT01360
 CAT01370
 CAT01380
 CAT01390
 CAT01400
 CAT01410
 CAT01420
 CAT01430
 CAT01440
 CAT01450
 CAT01460
 CAT01470
 CAT01480
 CAT01490
 CAT01500
 CAT01510

FILE CATGRY

```

C
C
C      FIND THE MAX. CATEGORY PDF (THIS PDF IS THE SUMMATION OF ALL
C                                THE PDF'S OF THE SUBCLASS WITHIN
C                                THE CATEGORY)
C
C      PK = 0.0
C      DO 175 LL=1,NOCAT
C      IF ( P(ILL) .LE. PK) GO TO 175
C      PK = P(ILL)
C      IC = LL
C 175  CONTINUE
C
C      ALL OF THE SUBCLASS PDF'S WERE TOO SMALL TO EXP., THEREFORE
C      THIS PIXEL WILL NOT BE CLASSIFIED.
C
C      IF (PK .NE. 0.0) GO TO 180
C      IR(II) = NOSUB2 + 1
C      VR(II) = 0.0
C      PTSTHS = PTSTHS + 1
C      GO TO 250
C 180  CONTINUE
C
C      STORE THE LARGEST SUBCLASS PDF AND SUBCLASS NO. OF THE CATEGORY
C      WITH THE MAX. PDF
C
C      IR(II) = SUBNUM(IC)
C      VR(II) = IFMAX(IC)
C 250  CONTINUE
C 350  CONTINUE
C      RETURN
C      END

```

CAT01520
 CAT01530
 CAT01540
 CAT01550
 CAT01560
 CAT01570
 CAT01580
 CAT01590
 CAT01600
 CAT01610
 CAT01620
 CAT01630
 CAT01640
 CAT01650
 CAT01660
 CAT01670
 CAT01680
 CAT01690
 CAT01700
 CAT01710
 CAT01720
 CAT01730
 CAT01740
 CAT01750
 CAT01760
 CAT01770
 CAT01780
 CAT01790
 CAT01800
 CAT01810
 CAT01820
 CAT01830

FILF: CATSCN

| | | |
|---|---|----------|
| C | FUNCTION CATSCN(CARD,KCLSNA,CATNME,KK,NOCLSS,NOCAT) | CAT00010 |
| | IMPLICIT INTEGER (A-Z) | CAT00020 |
| | DIMENSION KCLSNA(1),CARD(62),IBUFF(6) | CAT00030 |
| C | ----- | CAT00040 |
| C | CATSCN SCANS THE CATEGORY CARD FROM CLASSIFY AND STORES THE | CAT00050 |
| C | CATEGORY NAME IN CATNAM AND STORE THE CLASS NAMES IN KCLSNA | CAT00060 |
| C | ----- | CAT00070 |
| | DATA BLANK/' ',SLASH/'/',STAR/'*',COMMA/',', | CAT00080 |
| | LOGICAL*1 LCHAR(4),LLCHAR(4) | CAT00090 |
| | DIMENSION ICHAR(1),IICHAR(1) | CAT00100 |
| | EQUIVALENCE(LCHAR(1),ICHAR(1)),(LLCHAR(1),IICHAR(1)) | CAT00110 |
| | K = 1 | CAT00120 |
| | 6 COL = 0 | CAT00130 |
| A | KK = KK + 1 | CAT00140 |
| C | 10 J = NXTCHR(CARD,COL) | CAT00150 |
| | IF(J.EQ.BLANK)GO TO 110 | CAT00160 |
| | IF(J.EQ.COMMA)GO TO 10 | CAT00170 |
| | IF(J.EQ.SLASH)GO TO 10 | CAT00180 |
| | IF(J.EQ.STAR)GO TO 100 | CAT00190 |
| C | PICK CHARACTERS OFF CARD ONE AT A TIME | CAT00200 |
| C | LL = 1 | CAT00210 |
| | DO 20 I=1,6 | CAT00220 |
| | J2 = CARD(COL) | CAT00230 |
| | IF(J2.EQ.SLASH)GO TO 30 | CAT00240 |
| | IF(J2.EQ.STAR)GO TO 30 | CAT00250 |
| | IF(J2.EQ.COMMA)GO TO 30 | CAT00260 |
| | IF(J2.EQ.BLANK)GO TO 20 | CAT00270 |
| | IBUFF(LL) = J2 | CAT00280 |
| | LL = LL + 1 | CAT00290 |
| | COL = COL + 1 | CAT00300 |
| | 20 CONTINUE | CAT00310 |
| | 30 COL = COL - 1 | CAT00320 |
| | IF (LL.EQ. 7) GO TO 35 | CAT00330 |
| | GO TO 50 | CAT00340 |
| C | 35 DO 40 JJ=LL,6 | CAT00350 |
| | 40 IBUFF(JJ) = BLANK | CAT00360 |
| C | 50 CONTINUE | CAT00370 |
| | DO 60 I=1,4 | CAT00380 |
| | IICHAR(I)=IBUFF(I) | CAT00390 |
| | 60 LCHAR(I)=LLCHAR(I) | CAT00400 |
| | WRD1=ICHAR(1) | CAT00410 |
| C | GO TO (70,80),K | CAT00420 |
| C | WRD1 CONTAINS CATEGORY NAME | CAT00430 |
| C | 70 CATNME = WRD1 | CAT00440 |
| | K = 2 | CAT00450 |
| | GO TO 10 | CAT00460 |
| C | WRD1 CONTAINS CLASS NAME | CAT00470 |
| C | 80 KCLSNA(KK) = WRD1 | CAT00480 |
| | NOCLSS = NOCLSS + 1 | CAT00490 |
| | GO TO 8 | CAT00500 |
| C | NEXT CARD IS A CONTINUATION CARD | CAT00510 |
| C | 100 READ(21,500)CARD | CAT00520 |
| | COL = 0 | CAT00530 |
| | WRITE(6,550)CARD | CAT00540 |
| | 500 FORMAT(10X,62A1) | CAT00550 |
| | 550 FORMAT(115, 62A1) | CAT00560 |
| | GO TO 10 | CAT00570 |
| C | FINISHED SCANNING CARD | CAT00580 |
| C | 110 KK = KK - 1 | CAT00590 |
| | CATSCN = KK | CAT00600 |
| C | RETURN | CAT00610 |
| | END | CAT00620 |
| | | CAT00630 |
| | | CAT00640 |
| | | CAT00650 |
| | | CAT00660 |
| | | CAT00670 |
| | | CAT00680 |
| | | CAT00690 |
| | | CAT00700 |
| | | CAT00710 |
| | | CAT00720 |
| | | CAT00730 |
| | | CAT00740 |
| | | CAT00750 |
| | | CAT00760 |
| | | CAT00770 |
| | | CAT00780 |
| | | CAT00790 |

FILE: CLSFY1

SUBROUTINE CLSFY1(COVMTX,AVEMTX,FLDMTX,CLSMTX,APRIOR,
* BMATR,VERTEX,SUBDES,SUBNO,COVNEW,AVENEW,KATNO)

IMPLICIT INTEGER (A-H,O-Z)

CALL... CALL CLSFY1(COVMTX,AVEMTX,FLDMTX,RUNMTX,CLSMTX,APRIOR,
BMATR,COVNEW,AVENEW)

ARGS... COVMTX : LOCATION OF COVARIANCE MATRICES (SYMETTRIC
STORAGE) FOR NOCLS2 TRAINING CLASSES.
AVEMTX : LOCATION OF NOCLS2 TRAINING CLASS MEAN VECTORS
(NOFET2 MEANS PER CLASS)
FLDMTX : LOCATION OF TRAINING FIELD(S) INFORMATION
CLSMTX : LOCATION OF NAME FOR EACH CLASS
APRIOR : LOCATION OF APRIORI PROBABILITY VALUES FOR
EACH CLASS

VERTEX : LOCATION OF VERTICES OF SAVED TRAINING FIELDS

SUBDES : LOCATION OF SUBCLASS NAMES

SUBNO : LOCATION OF ARRAY CONTAINING NO. OF SUBCLASSES IN
EACH CLASS

COVNEW : LOCATION USED TO STORE 'B'-TRANSFORMED
COVARIANCE MATRICES.

CONTINUE

AVENEW : LOCATION USED TO STORE THE 'B'-TRANSFORMED
MEAN VECTORS.

BMATR : LOCATION OF THE 'B'-TRANSFORMATION MATRIX,
IF AVAILABLE, FOR APPLICATION TO THE CLASS
MEANS AND COVARIANCE MATRICES.

KATNO : CATEGORY - CLASS CORRESPONDENCE

PURPOSE...

IF AVAILABLE, THE 'B'-TRANSFORMATION MATRIX IS APPLIED
TO THE SUBCLASS MEAN VECTORS AND COVARIANCE MATRICES.
OBTAINS THE (MODIFIED) CHOLESKY FACTORIZATION OF THE
SUBCLASS COVARIANCE MATRICES, PROVIDES THE 'CONSTANT' OF
THE PROBABILITY DENSITY FUNCTION AND DETERMINANT FOR
EACH SUBCLASS, AND OBTAINS THE SUBCLASS-PAIR THRESHOLDS FOR
USE BY SUBR. CONTEX IN CLASSIFICATION OF INPUT SCAN
LINES. PUBLISHES AND OUTPUTS ON MAPTAP THE TRAINING
FIELD(S) INFORMATION AND THE STATISTICS FOR EACH OF THE
TRAINING CLASSES.

RETURNS...CHOLESKY FACTORIZATION OF THE INPUT COVARIANCE MATRICES
(AFTER 'B'-TRANSFORMATION, IF APPLICABLE), SUBCLASSPAIR
THRESHOLDS, AND SUBCLASS STATISTICS OUTPUT ON MAPTAP.
CONTINUE

INCLUDE COMPK1.LIST

INCLUDE COMPK2.LIST

* COMMON/INFORM/NOCLS2,NOSUB2,NOFET2,VARS22,TOTVT2,NOFLD2,
* AVAR2,COVAR2,CLSID2,SURN02,SURDS2,FLDSV2,VERTX2,
* FFTVC2(30),SUHVC2(75),SUBPTR(75),CLSV2(60),

CLS00010
CLS00020
CLS00030
CLS00040
CLS00050
CLS00060
CLS00070
CLS00080
CLS00090
CLS00100
CLS00110
CLS00120
CLS00130
CLS00140
CLS00150
CLS00160
CLS00170
CLS00180
CLS00190
CLS00200
CLS00210
CLS00220
CLS00230
CLS00240
CLS00250
CLS00260
CLS00270
CLS00280
CLS00290
CLS00300
CLS00310
CLS00320
CLS00330
CLS00340
CLS00350
CLS00360
CLS00370
CLS00380
CLS00390
CLS00400
CLS00410
CLS00420
CLS00430
CLS00440
CLS00450
CLS00460
CLS00470
CLS00480
CLS00490
CLS00500
CLS00510
CLS00520
CLS00530
CLS00540
CLS00550
CLS00560
CLS00570
CLS00580
CLS00590
CLS00600
CLS00610
CLS00620
CLS00630
CLS00640
CLS00650
CLS00660
CLS00670
CLS00680
CLS00690
CLS00700
CLS00710
CLS00720
CLS00730
CLS00740
CLS00750
CLS00760
CLS00770
CLS00780
CLS00790

CSEND

[illegible]

FILE: CLSFY1

```
C
65 FORMAT(3X, '*** CLASSIFICATION STUDY *** MAPTAP FILE ', 5X, I2, '//')
10 CALL WRTFLD(FLOMTX, VERTEX, NOFLD2, CLSMTX, SURDES)

PUBLISH THE CLASSES AND CHANNELS (WITH SPECTRAL BAND) TO BE
CONSIDERED IN CLASSIFICATION.

WRITE(6, HEAD)
WRITE(6, 65) NFILE
WRITE(6, 20)
20 FORMAT( '////T27, SUBCLASSES CONSIDERED', T90, 'CHANNELS CONSIDERED',
*//T21, SYMBOL, T32, SUBCLASS, T45, A PRIOR, T88, TRAINING RECOGN
*ITION')
II = NOSUR2
IF ( II .LT. NOFET2 ) II = NOFET2
DO 30 I=1, II
WRITE(6, 40)
IF ( I .LE. NOSUR2 ) WRITE(6, 50) CLSSYM(I), SURDES(I), APRIOR(I)
IF ( I .LE. NOFET2 ) WRITE(6, 60) FETVC2(I), CHNVEC(I)
30 CONTINUE
40 FORMAT( ' ', I )
50 FORMAT( ' ', T23, A1, T33, A4, T45, F7.4 )
60 FORMAT( ' ', T91, I2, T103, I2 )

SAVE AND PUBLISH THE MEAN AND COVARIANCE
-----

IF R-MATRIX IS AVAILABLE. TRANSFORM THE COVARIANCE MATRIX AND
MEAN VECTOR

TEMPE2 = NOFET2
IF ( RMFLG .LE. 0 ) GO TO 611
DO 610 NCLS=1, NOSUR2
DO 605 K=1, RMCOMB
RRM102 = ( R * (K-1) ) / 2
DO 603 C=1, R
I = RRM102 + C
SUMTR = 0.0
DO 602 J=1, NOFET2
SUM = 0.0
DO 601 K=1, NOFET2
II = ( K * (K-1) ) / 2 + J
IF ( J .GT. K ) II = ( J * (J-1) ) / 2 + K
601 SUM = SUM + RMATR(R, K) * COVMTX(II, NCLS)
602 SUMTR = SUMTR + RMATR(C, J) * SUM
603 RXXKRT(I) = SUMTR
SUM = 0.0
DO 604 J=1, NOFET2
SUM = SUM + RMATR(R, J) * AVEMTX(J, NCLS)
604 VEC(R) = SUM
605 CONTINUE
DO 606 J=1, I
COVNEW(J, NCLS) = RXXKRT(J)
DO 607 J=1, RMCOMB
607 AVENFW(J, NCLS) = VEC(J)
610 CONTINUE
VAR572 = RMFLG
NOFET2 = RMCOMB
611 CONTINUE
GO TO R0
70 CONTINUE

OBTAIN THE SUBCLASS-PAIR THRESHOLDS, FOR USE BY CLSFY2/CONTEX

IF (NOCAT .GT. 0) GO TO 612
NPL1 = NOFET2 + 1
CALL THRESH(NOSUR2, NOFET2, NPL1, APRIOR, AVEMTX, COVMTX, DET, VAR572,
1 SCH2(1), SCH2(901), SCH2(1801), SCH2(1831), SCH2(1861),
2 SCH2(2791), SCH1(TH1J))
612 CONTINUE

NOFET2 = TEMPE2
RETURN
R0 III = 0
```

FILE: CLSFY1

```
-----
OUTPUT THE ORIGINAL COVARIANCE AND MEAN MATRIX( AFTER B-TRANSFORM
IF B-MATRIX AVAILABLE) FOR ALL SUBCLASSES, ON THE CLASSIFICATION
RESULTS OUTPUT FILE, MAPTAP
-----

HEADER RECORD NO. 3 FOR MAPTAP
WRITE(MAPTAP) ((COVMTX(I,J),I=1,VAR5Z2),J=1,NOSUB2),
  ((AVEMTX(I,J),I=1,NOFET2),J=1,NOSUB2)
BADFLG = 0

IF (STATKY.F0.0) GO TO 180
CNT = 7*(5+3*2*NOFET2)*((NOFET2+11)/12)
CNT = PAGESIZ/CNT
INC = CNT
DO 170 ICLAS=1,NOCLS2
  NUMSUB = SURNO(ICLAS)
  DO 120 I=1,NUMSUB
    III = III + 1
    IF (INC.LT.CNT) GO TO 100
    WRITE(6,HEAD)
    WRITE(6,65)NFILE
    INC = 0
100 CONTINUE
    WRITE(6,110) CLSMTX(ICLAS),SUBDES(III)
110 FORMAT(/// CLASS :',A4/' SUBCLASS: ',A4)
    DO 140 LOC=1,NOFET2,12
      STOP = LOC+11
      IF( STOP.GT. NOFET2 ) STOP = NOFET2
140 WRITE(6,150) ( AVEMTX(I,III),I=LOC,STOP)
150 FORMAT(10 MEAN:',3X,12F9.2)

IF( RMFLG .GT. 0) GO TO 161
WRITE(6,160)
160 FORMAT(/// COVARIANCE MATRIX: )
GO TO 162
161 WRITE(6,1611)
1611 FORMAT(/// COVARIANCE MATRIX (B-TRANSFORMED) : )
162 CALL WRTMTX(COVMTX(1,III),NOFET2, BCDTWO)
INC = INC+1
120 CONTINUE
170 CONTINUE

OBTAIN THE 'MODIFIED' CHOLESKY DECOMPOSITION OF THE COVARIANCE
MATRIX FOR EACH SUBCLASS, THE DETERMINANT, AND COMPUTE THE
'CONSTANT' TERM OF THE PROBABILITY DENSITY FUNCTION
= -2 * LOG Q(I) + LOG DETERMINANT(I), WHERE Q(I) IS THE
APRIORI PROBABILITY VALUE FOR SUBCLASS I

PDF = Q(I) * ( DET(I)**-1/2 * E** -1/2 * (X-M) * K** -1 * (X-M) )
LOG PDF = -1/2 * ( CON + (X-M) * K** -1 * (X-M) )

180 DO 195 NCLS=1,NOSUR2
  TRANSFER ORIGINAL COVARIANCE MATRIX TO TEMPORARY STORAGE ( COV )
  DO 181 I=1,VAR5Z2
    COV(I) = COVMTX(I,NCLS)
181
  OBTAIN THE 'MODIFIED' CHOLESKY FACTORIZATION OF THE
  COVARIANCE MATRIX
  CALL MCHLSK( COVMTX(1,NCLS), NOFET2, DUM, DET(NCLS) )

  IF( DET(NCLS) .GT. 0.0) GO TO 183
  WRITE(6,1800) NCLS, DET(NCLS)
1800 FORMAT(/// 5X, '***** CLSFY/CLSFY1/----- THE COVARIANCE MAT
  RIX FOR SUBCLASS NO',I4,' IS EITHER SINGULAR OR NOT POSITIVE DEFIN
  CLS02380
  CLS02390
  CLS02400
  CLS02410
  CLS02420
  CLS02430
  CLS02440
  CLS02450
  CLS02460
  CLS02470
  CLS02480
  CLS02490
  CLS02500
  CLS02510
  CLS02520
  CLS02530
  CLS02540
  CLS02550
  CLS02560
  CLS02570
  CLS02580
  CLS02590
  CLS02600
  CLS02610
  CLS02620
  CLS02630
  CLS02640
  CLS02650
  CLS02660
  CLS02670
  CLS02680
  CLS02690
  CLS02700
  CLS02710
  CLS02720
  CLS02730
  CLS02740
  CLS02750
  CLS02760
  CLS02770
  CLS02780
  CLS02790
  CLS02800
  CLS02810
  CLS02820
  CLS02830
  CLS02840
  CLS02850
  CLS02860
  CLS02870
  CLS02880
  CLS02890
  CLS02900
  CLS02910
  CLS02920
  CLS02930
  CLS02940
  CLS02950
  CLS02960
  CLS02970
  CLS02980
  CLS02990
  CLS03000
  CLS03010
  CLS03020
  CLS03030
  CLS03040
  CLS03050
  CLS03060
  CLS03070
  CLS03080
  CLS03090
  CLS03100
  CLS03110
  CLS03120
  CLS03130
  CLS03140
  CLS03150
  CLS03160
```

FILE1 CLSFY1

```

2ITE1// 35X,DETERMINANT =,F20.4///5X,***** TERMINATING PROGRAM CLS03170
3 EXECUTION ***** ) CLS03180
C RADELG = 99 CLS03190
GO TO 195 CLS03200
183 APR = APRIOR(NCLS)**(-2) CLS03210
CON(NCLS) = ALOG( APR * DET(NCLS) ) CLS03220
CLS03230
OBTAIN THE RELATIVE ERROR OF FACTORED MATRIX ( RATIO OF CLS03240
EUCLIDEAN NORM OF DIFFERENCE, K-LDL* , TO EUCLIDEAN NORM OF K) CLS03250
ERROR(NCLS) = RELERR( COVMTX(1,NCLS),COV,NOFET2,VARSZ2) CLS03260
CLS03270
195 CONTINUE CLS03280
ERROR TERMINATION, IF ONE OR MORE SUBCLASSES HAVE AN INVALID CLS03290
(SINGULAR, OR NON-POSITIVE DEFINITE) COVARIANCE MATRIX CLS03300
IF( BADFLG .GT. 0) CALL EXIT CLS03310
CLS03320
----- CLS03330
OUTPUT THE ( MODIFIED ) CHOLESKY FACTORIZATION OF THE CLS03340
COVARIANCE MATRIX, ON THE CLASSIFICATION OUTPUT FILE, MAPTAP CLS03350
----- CLS03360
CLS03370
HEADER RECORD NO. 4 FOR MAPTAP CLS03380
WRITE(MAPTAP) ((COVMTX(I,J),I=1,VARSZ2),J=1,NOSUB2), CLS03390
(CON(I),I=1,NOSUB2),(DET(I),I=1,NOSUB2) CLS03400
IF (STATKY.F0.0) GO TO 230 CLS03410
CNT = 13 * (5*2*NOFET2) * ((NOFET2+11)/12) CLS03420
CNT = PAGES17/CNT CLS03430
INC = CNT CLS03440
C CLS03450
III = 0 CLS03460
DO 210 L=1,NOCLS2 CLS03470
NSUBCL = SURNO(L) CLS03480
DO 210 I=1,NSURCL CLS03490
III = III + 1 CLS03500
IF (INC.LT.CNT) GO TO 200 CLS03510
WRITE(6,HEAD) CLS03520
WRITE(6.65)NFILE CLS03530
INC = 0 CLS03540
200 WRITE(4.220)CLSMTX(L),L,SURDES(III),III,DET(III),CON(III) CLS03550
CALL WRITMX(COVMTX(1,III),NOFET2,RCDFOR) CLS03560
C CLS03570
WRITE(6.205)ERROR(III) CLS03580
205 FORMAT(1X,10* RELATIVE ERROR ( EUCLIDEAN NORM (K-LDL*)/EUCLIDEAN CLS03590
INORM K ) =, F15.8 // ) CLS03600
C CLS03610
INC = INC+1 CLS03620
210 CONTINUE CLS03630
220 FORMAT(1H0// T50,1MULTISPECTRAL CHARACTERISTICS FOR1/T57,A4, CLS03640
1' ( CLASS1,13,1 )',/T56,A4,2X,1 ( SUBCLASS1,13,1 )'// CLS03650
2 1H0,DETERMINANT =,F25.4 / 1H0,1PROR. DENSITY FUCLS03660
2NCTION - CONSTANT TERM=,F10.4// 1H0,1COVARIANCE MATRIX (CHOLESKY CLS03670
3 FACTORIZATION) : / ) CLS03680
C CLS03690
C CLS03700
230 CONTINUE CLS03710
GO TO 70 CLS03720
END CLS03730
CLS03740
CLS03750
CLS03760
CLS03770
CLS03780
CLS03790
CLS03800
CLS03810
CLS03820
CLS03830
CLS03840

```

FILE: CLSFY2

SUBROUTINE CLSFY2(COVMTX,AVEMTX,FLOMTX,CLSMTX,SUBDES,SUBNO,
KATNO,BMATRX)

IMPLICIT INTEGER (A-M,O-Z)

CALL... CALL CLSFY2(COVMTX,AVEMTX,FLOMTX,RUNMTX,CLSMTX,BMATRX)
 ARGS... COVMTX : LOCATION OF THE COVARIANCE MATRICES
 (IN 'SYMETTRIC' STORAGE) FOR NOCLS2 TRAINING
 CLASSES
 AVEMTX : LOCATION OF THE MATRIX OF TRAINING CLASS MEAN
 VECTORS (NOFET2 MEANS PER CLASS)
 FLOMTX : LOCATION OF MATRIX OF TRAINING FIELD(S)
 INFORMATION
 CLSMTX : LOCATION OF MATRIX OF TRAINING CLASS NAMES
 BMATRX : B-TRANSFORMATION MATRIX,IF AVAILABLE
 SUBDES : LOCATION OF SUBCLASS NAMES
 SUBNO : LOCATION OF NO. OF SUBCLASSES IN EACH CLASS
 KATNO : CATEGORY - CLASS CORRESPONDENCE ARRAY
 BMATRX : LOCATION OF THE 'B'-TRANSFORMATION MATRIX,
 IF AVAILABLE, FOR TRANSFORMATION OF INPUT
 SAMPLE VECTOR IN SUBR. CONTEX

PURPOSE... CLASSIFIES THE SET OF SAMPLES (MULTI-CHANNEL DATA
 POINTS) ON EACH SCAN LINE OF THE SET OF SCAN LINES
 PRESCRIBED BY THE 'FIELD DEFINITION' CARD INPUT TO
 THE CLASSIFICATION PROCESSOR. CLASSIFICATION IS
 PERFORMED BY THE METHOD OF MAXIMUM LIKELIHOOD
 (MINIMUM PROBABILITY OF MIS-CLASSIFICATION) IN
 SUBROUTINE CONTEX. THE DIMENSIONALITY OF THE SAMPLE
 IS PRESCRIBED BY CONTROL CARD INPUT TO THE PROCESSOR
 (BMCOMR) OF CHANNELS (BMFEAT) IN THE 'B' - MATRIX,
 ('CHANNELS') OR BY THE NO. OF LINEAR COMBINATIONS
 IF AVAILABLE.

RETURNS... 1. IF STANDARD CLASSIFIER IS USED, THE SUBCLASS
 NUMBER AND PROBABILITY DENSITY FUNCTION
 VALUE FOR EACH POINT OF EVERY SCAN LINE OF THE FIELD
 IS OUTPUT ON THE CLASSIFICATION OUTPUT FILE, MAPTAP.
 2. IF CATEGORY CLASSIFIER IS USED, THE SUBCLASS
 NUMBER OF THE SUBCLASS WITH THE LARGEST PROBABILITY
 DENSITY FUNCTION WITHIN THE CHOSEN CATEGORY AND THE
 PROBABILITY DENSITY FUNCTION VALUE OF THE CHOSEN
 CATEGORY FOR EACH POINT OF EVERY SCAN LINE OF THE
 FIELD IS OUTPUT ON THE CLASSIFICATION OUTPUT FILE.

INCLUDE COMHKA.LIST

INCLUDE COMHK1.LIST

COMMON/INFO-PM/NOCLS2,NOSUB2,NOFET2,VARSZ2,TOTVT2,NOFLD2,
 AVAR2,COVAR2,CLS1D2,SUBNO2,SUBDS2,FLDSV2,VERTX2,
 FETVC2(30),SUHVC2(75),SURPTH(75),CLSVC2(60),
 KEPTS(60),NOGMP,GMPNAM(60),GMPDFX(61),
 GRPCHK(61),GROUPS(124)
 COMMON /CLASS/ APRFLG,HMCOMH,HMFEAT,HMFLG,NOCAT,THIJ1,DATA1,
 NFILE,STATKY,CATNAM(60),
 CLSSYM(60),CON(60),DET(60),FLDESC,FLDINF(6),
 KCLSN(60),NOCTCL(60),SURCAT(60)
 ,NOCHAN,CHNVEC(30)
 COMMON/GLUBAL/HEAD(63),MAPTAP,DATAPE,SAVTAP,HMFILE,BMKEY.

CLS00010
 CLS00020
 CLS00030
 CLS00040
 CLS00050
 CLS00060
 CLS00070
 CLS00080
 CLS00090
 CLS00100
 CLS00110
 CLS00120
 CLS00130
 CLS00140
 CLS00150
 CLS00160
 CLS00170
 CLS00180
 CLS00190
 CLS00200
 CLS00210
 CLS00220
 CLS00230
 CLS00240
 CLS00250
 CLS00260
 CLS00270
 CLS00280
 CLS00290
 CLS00300
 CLS00310
 CLS00320
 CLS00330
 CLS00340
 CLS00350
 CLS00360
 CLS00370
 CLS00380
 CLS00390
 CLS00400
 CLS00410
 CLS00420
 CLS00430
 CLS00440
 CLS00450
 CLS00460
 CLS00470
 CLS00480
 CLS00490
 CLS00500
 CLS00510
 CLS00520
 CLS00530
 CLS00540
 CLS00550
 CLS00560
 CLS00570
 CLS00580
 CLS00590
 CLS00600
 CLS00610
 CLS00620
 CLS00630
 CLS00640
 CLS00650
 CLS00660
 CLS00670
 CLS00680
 CLS00690
 CLS00700
 CLS00710
 CLS00720
 CLS00730
 CLS00740
 CLS00750
 CLS00760
 CLS00770
 CLS00780
 CLS00790

FILE: CLSPY2

```

      HISFIL,HISKEY,TRFORM,ERPTP,ERPKEY,MAPUNT,NOFILE,
      DRUMAD,DRMWS,PAGSIZ,DATFIL,STAFIL,ASAV,ASAVFL
      NMSTUN,NMSTFI,SCRUN,MAPFIL
      DOTUNT,DOTFIL,NCHPAS,TRNSFL,BMTRFL,HISTFL,PCHUNT,
      CHDUNT,PTUNT,RANDIO
CSEND
      REAL CON,DET,      VR(1000),COVMTX(VARSZ2,NOSUB2),AVEMTX(NOFET2,
      NOSUB2),BMATRIX

      DIMENSION FLOMTX(4,NOFLD2),CLSMTX(1),VERTCS(22)
      1 COL(3,110),OUT(110),IR(1000)
      2 SURNO(1),SURDES(1),KATNO(1)

      *** NOTE : THE IDATA ARRAY IS USED FOR INPUT OF THE SCAN LINE TO
      BE CLASSIFIED TO SURR. CONTEX. AND ALSO TO STORE THE
      CLASSIFIED SCAN LINE, BY CONTEX. THE 'IR' AND 'VR'
      ARRAYS, EQUIVALENCED TO THE IDATA ARRAY, ARE
      LOCATIONS USED BY CONTEX TO OUTPUT THE CLASSIFIED
      SCAN LINE AND THE PROBABILITY DENSITY FUNCTION
      VALUES, RESPECTIVELY.

      -----
      INCLUDE COMPK2.LIST
      EQUIVALENCE      (FLOINF(1),LINSTR), (FLOINF(2),LINEND),
      1 (FLOINF(3),LININC), (FLOINF(4),SAMSTR),
      2 (FLOINF(5),SAMEND), (FLOINF(6),SAMINC),
      3 (FLOINF(7),FLDTYP)

      -----
      COMMON /SCRACH/ IDATA(12500)

      -----
      DATA LINMAX/1000/, ENDBCD/'SEND'//, DASH/'-----'/

      EQUIVALENCE (IR,COL)
      DATA LEWN/'( '//'BLANK/' '

      -----
      READ THE FIELD DESCRIPTION CARD,CONTAINING LINE-SAMPLE COORDINATES
      OF THE FIELD TO BE CLASSIFIED
      -----
      CALL TAPHOR(DATAPE,DATFIL)

      10 CONTINUE
      PSTMS = 0
      ICK = LAHEAD(FLDESC,VERTCS,FLOINF,NC)
      IF ( ICK .EQ. 0 ) GO TO 160
      IF (ICK .EQ. -1) .OR. ICK .EQ. -2) GO TO 10
      30 CONTINUE

```

CLS00800
 CLS00810
 CLS00820
 CLS00830
 CLS00840
 CLS00850
 CLS00860
 CLS00870
 CLS00880
 CLS00890
 CLS00900
 CLS00910
 CLS00920
 CLS00930
 CLS00940
 CLS00950
 CLS00960
 CLS00970
 CLS00980
 CLS00990
 CLS01000
 CLS01010
 CLS01020
 CLS01030
 CLS01040
 CLS01050
 CLS01060
 CLS01070
 CLS01080
 CLS01090
 CLS01100
 CLS01110
 CLS01120
 CLS01130
 CLS01140
 CLS01150
 CLS01160
 CLS01170
 CLS01180
 CLS01190
 CLS01200
 CLS01210
 CLS01220
 CLS01230
 CLS01240
 CLS01250
 CLS01260
 CLS01270
 CLS01280
 CLS01290
 CLS01300
 CLS01310
 CLS01320
 CLS01330
 CLS01340
 CLS01350
 CLS01360
 CLS01370
 CLS01380
 CLS01390
 CLS01400
 CLS01410
 CLS01420
 CLS01430
 CLS01440
 CLS01450
 CLS01460
 CLS01470
 CLS01480
 CLS01490
 CLS01500
 CLS01510
 CLS01520
 CLS01530
 CLS01540
 CLS01550
 CLS01560
 CLS01570
 CLS01580

ORIGINAL PAGE IS
 OF POOR QUALITY

11-13

149

```

WRITE(6,HEAD)
NC1 = 2*NC
NV = NC - 1
WRITE(6,35)FLDESC,NV,FLDINF(6),FLDINF(3),(LPRN,VERTCS(I),
* VERTCS(I+1)),I=1,NC1,2)
37 FORMAT(/// 156,'AREA OF CLASSIFICATION'/// 36X,'SAMPLE',3X,
1 'LINE', / 5X,'FIELD NAME',3X,'NO. OF VERTICES',3X,'INC. ',
2 3X,'INC.',30X,'VERTICES' / 7X,A4,14X,12,10X,12,6X,12,5X,
3 5(A1,14,'.',14,''),2X) / (52X,5(A1,14,'.',14,''),2X)/))
SAVE CLASSIFIED FIELD INFORMATION ON MAPTAP
-----
50 CONTINUE
INITIALIZE TAPE READING FOR THIS FIELD
CALL FLDINT(FLDINF,CHNVEC,NOCHAN)

LINES = ( LINEND-LINSTR)/LININC + 1
PTS = ( SAMEND-SAMSTR)/SAMINC + 1
IF (PTS .GT. LINMAX) WRITE(6,55)
55 FORMAT( // 5X,'WIDTH OF RECTANGULAR FIELD SURROUNDING CLASSIFICATION'
* ON FIELD CANNOT EXCEED 1000 POINTS.')
IF (PTS .GT. LINMAX) CALL CMERR

SCANLN = PTS * NOFET2
IF (SCANLN .GT. (12500-IDATA1+1)) GO TO 170

FIELD RECORD FOR MAPTAP
WRITE(MAPTAP)(FLDINF(I),I=1,6),PTS,LINES,FLDESC,NC,
* (VERTCS(I),I=1,NC),(VERTCS(I+NC),I=1,NC)

CLASSIFY THE FIELD
-----
CALL SETMRG(66,0,66)

WRITE(6,HEAD)
CALL MAPMDG(NOCAT,CLSSYM,CATNAM,KATNO,CLSMTX,SUBNO,SUBDES)
WRITE(6,R00)
800 FORMAT(///)
J = 0
DO 80 I=SAMSTR,SAMEND,SAMINC
J = J+1
COL(1,J) = I/100
COL(2,J) = MOD(I,100)/10
COL(3,J)=MOD(I,10)
IF (J.EQ.110) GO TO 90
80 CONTINUE
90 DO 100 I=1,3
100 WRITE(6,110)(COL(I,K),K=1,J)
110 FORMAT(' ',9X,110I1)
WRITE(6,115)
115 FORMAT( / )

ILINE=LINSTR-LININC
DO 140 JLINE=1,LINES
CALL LINEWD(IDATA(IDATA1),ENDTAP)
IF ( ENDTAP .EQ. -1) GO TO 150
ILINE=ILINE+LININC

*** CLASSIFY THE SCAN LINE IN IDATA

CALL STANDARD CLASSIFIER

IF (NOCAT .LE. 0) CALL CONTEXT(NOFET2,NOSUB2,PTS,AVEMTX,COVMTX,
* BMATRX,IDATA(IDATA1),VERTCS,NC,IR,VR,ILINE,IDATA(THIJI))

CALL CATEGORY CLASSIFIER

IF (NOCAT .GT. 0) CALL CATGRY(NOFET2, PTS,AVEMTX,COVMTX,
* IR,VR,HMATRX,IDATA(IDATA1),ILINE,VERTCS,NC,PTSIHS)

```

FILE: CLSFY2

```
C
DO 120 JKL=1,J                                CLS02380
ISYMB = IR(JKL)                                CLS02390
IF ( ISYMB .EQ. 0 ) OUT(JKL) = BLANK            CLS02400
IF ( ISYMB .NE. 0 ) OUT(JKL) = CLSSYM(ISYMB)    CLS02410
120 CONTINUE                                    CLS02420
WRITE (6,130) ILINE,(OUT(I),I=1,J)             CLS02430
130 FORMAT(' ',15,4X,110A1)                    CLS02440
C                                                CLS02450
C SAVE CLASSIFIED INFORMATION ON MAPTAP          CLS02460
C ----- CLS02470
C DATA RECORD FOR MAPTAP                      CLS02480
C
C WRITE(MAPTAP) ILINE,(IR(I),I=1,PTS),(VR(I),I=1,PTS) CLS02490
140 CONTINUE                                    CLS02500
IF (PTSTHS .GT. 0) WRITE(6,145) PTSTHS          CLS02510
145 FORMAT(///9X,'AS THE COMPUTER CANNOT EXPONENTIATE A NUMBER SMALLER CLS02520
      *R THAN EXP(-88),'',16,' PTS WERE NOT CLASSIFIED IN THIS FIELD') CLS02530
150 ILINE = 0                                    CLS02540
C END OF FIELD RECORD FOR MAPTAP                CLS02550
C
C WRITE(MAPTAP) ILINE,(IR(I),I=1,PTS),(VR(I),I=1,PTS) CLS02560
C
C GO TO 10                                       CLS02570
C
C GO HOME                                        CLS02580
C ----- CLS02590
C
C 160 PTS = 0                                    CLS02600
C
C END OF RUN RECORD FOR MAPTAP                  CLS02610
C
C WRITE(MAPTAP) (FLDINF(I),I=1,6),PTS,LINES,FLDESC,NC, CLS02620
      (VERTCS(I),I=1,NC),(VERTCS(I+NC),I=1,NC) CLS02630
C
C CALL SFTMRG(66,4,62)                         CLS02640
C
C RETURN                                         CLS02650
C ----- CLS02660
C
C 170 WRITE(6,175)                              CLS02670
175 FORMAT(//5X,'TOO MUCH DATA REQUESTED. DO ONE OF THE FOLLOWING:' CLS02680
      *//7X,'1) FOR STANDARD CLASSIFIER -//11X,'REDUCE PARAMETERS SUCH TH CLS02690
      *AT//11X,'(NO. OF SUBCLASSES-1)*(NO. OF SUBCLASSES-2)/2 + NO. OF SU CLS02700
      *RCLASSES + (PTS PER SCAN LINE)*(NO. OF CHANNELS) + 12500'//7X, CLS02710
      *//2) FOR CATEGORY CLASSIFIER -//11X,'REDUCE DATA REQUESTED SUCH T CLS02720
      *HAT //11X,'(PTS PER SCAN LINE)*(NO. OF CHANNELS) + 12500.'') CLS02730
      CALL CMERR CLS02740
      END CLS02750
C CLS02760
C CLS02770
C CLS02780
C CLS02790
C CLS02800
C CLS02810
C CLS02820
C CLS02830
C CLS02840
C CLS02850
C CLS02860
C CLS02870
```

[illegible]

```

INCLUDE COMRK2,LIST
COMMON /CLASS/ APRFLG,BMCOMB,BMFEAT,BMFLG,NOCAT,THIJ1,IDATA1,
*          NFILE,STATKY,CATNAM(60),
3          CLSSYM(60),CON(60),DET(60),FLDESC,FLDINF(6),
4          KCL,SNA(60),NOCTCL(60),SUBCAT(60)
*          ,NOCHAN,CHNVEC(30)

```

LOGICAL .BMFLAG , KOI

DIMENSION TH(1)

THE PRE-COMPUTED CLASS-PAIR THRESHOLDS, IN TH, ARE USED TO MINIMIZE THE NUMBER OF CLASS PROBABILITY DENSITY FUNCTIONS (PDF) COMPUTED TO OBTAIN THE MAXIMUM PDF FOR A GIVEN SAMPLE BEING CLASSIFIED.

```
CALL... CALL CONTEX(NCHAN,NC,NPTS,AVE,COV,BMATR,IDATA,VERTCS,  
VT,IR,VR,ILINE)
```

NC : THE NUMBER OF SUBCLASSES (TRAINING CLASSES. CONTINUE

FOR WHICH COVARIANCE MATRICES, MEAN VECTORS,
AND CLASS-PAIR THRESHOLDS ARE AVAILABLE -
REPRESENTS THE MAXIMUM NUMBER OF POSSIBILITIES
FOR CLASSIFICATION OF EACH DATA SAMPLE.)

NPTS : THE NUMBER OF INPUT DATA POINTS (PER CHANNEL)
ON THE RECTANGULAR FIELD

VERTICS : VERTICES OF FIELD TO BE CLASSIFIED

CON00010
CON00020
CON00030
CON00040
-CON00050
CON00060
CON00070
CON00080
CON00090
CON00100
CON00110
CON00120
CON00130
CON00140
CON00150
CON00160
CON00170
CON00180
CON00190
-CON00200
CON00210
CON00220
-CON00230
CON00240
-CON00250
CON00260
CON00270
CON00280
CON00290
CON00300
CON00310
CON00320
CON00330
CON00340
-CON00350
CON00360
CON00370
ECON00380
CON00390
INCON00400
CON00410
CON00420
CON00430
CON00440
CON00450
CON00460
CON00470
CON00480
CON00490
CON00500
CON00510
CON00520
I CON00530
I CON00540
I CON00550
CON00560
CON00570
CON00580
CON00590
CON00600
CON00610
CON00620
CON00630
CON00640
CON00650
CON00660
CON00670
CON00680
I CON00690
CON00700
CON00710
CON00720
CON00730

FILE: CONTEX

```

C      VT : NO. OF VERTICES OF FIELD TO BE CLASSIFIED          CON00740
C      IR : WILL CONTAIN THE CLASSIFIED DATA                   CON00750
C      VR : WILL CONTAIN THE CORRESPONDING PDF OF IR ARRAY      CON00760
C                                                                CON00770
C      INLINE : SCAN LINE NUMBER FROM DATA TAPE                CON00780
C                   TO BE CLASSIFIED ON EACH INPUT SCAN LINE.   CON00790
C                                                                CON00800
C      AVE : THE VECTOR OF 'NCHAN' MEANS( OR 'BMCOMB'          CON00810
C            MEANS IF THE 'B-TRANSFORMATION' MATRIX            CON00820
C            HAS BEEN APPLIED).                                  CON00830
C                                                                CON00840
C      COV : COVARIANCE MATRIX ( AFTER CHOLESKY                CON00850
C            FACTORIZATION) - ACTUAL DIMENSION OF COV. MATRIX   CON00860
C            IS DEPENDENT ON WHETHER IT HAS BEEN                CON00870
C            'B-TRANSFORMED'.                                    CON00880
C                                                                CON00890
C      CONTINUE                                                  CON00900
C      CON : THE 'CONSTANT' OF THE PROB. DENSITY FUNCTION      CON00910
C            FOR EACH CLASS                                     CON00920
C                                                                CON00930
C      BMATR : THE 'B' -TRANSFORMATION MATRIX                  CON00940
C                                                                CON00950
C      IDATA : THE SCAN LINE TO BE CLASSIFIED                  CON00960
C                                                                CON00970
C                                                                CON00980
C      OTHER...                                                 CON00990
C                                                                CON01000
C                                                                CON01010
C      BMFLG : A FLAG TO INDICATE THE PRESENCE OR ABSENCE OF   CON01020
C            'B-TRANSFORMATION' MATRIX .                       CON01030
C                                                                CON01040
C      BMCOMB : THE NUMBER OF LINEAR COMBINATIONS OF BMFEAT    CON01050
C            CHANNELS, IN THE 'B'-MATRIX.                      CON01060
C                                                                CON01070
C      -----I-----CON01080
C      -----I-----CON01090
C      -----I-----CON01100
C      EQUIVALENCE (FLDINF(1),LINSTR) , (FLDINF(2),LINEND) ,    CON01110
C      *             (FLDINF(3),LININC) , (FLDINF(4),SAMSTR) ,    CON01120
C      *             (FLDINF(5),SAMEND) , (FLDINF(6),SAMINC) )    CON01130
C                                                                CON01140
C                                                                CON01150
C      BMFLAG = BMFLG .GT. 0                                     CON01160
C      IF( BMFLAG ) GO TO 1000                                    CON01170
C      NV = NCHAN                                                 CON01180
C      GO TO 1001                                                  CON01190
C 1000 NV = BMCOMB                                                 CON01200
C 1001 K01 = NV .EQ. 1                                             CON01210
C                                                                CON01220
C      NPC = ( NV * (NV+1))/2                                       CON01230
C                                                                CON01240
C      NM1 = NC - 1                                                 CON01250
C      DO 1 I=1,NC                                                  CON01260
C 1      JOPOER(I) = I                                             CON01270
C                                                                CON01280
C      IPT1 = 1                                                     CON01290
C                                                                CON01300
C      2 JJ = IPT1                                                  CON01310
C      DO 100 I=1,NC                                                CON01320
C 100  NCNT(I) = 0                                                  CON01330
C                                                                CON01340
C      DO 110 I=1,NPTS                                              CON01350
C      IR(I) = 0                                                  CON01360
C 110  VR(I) = 0.0                                                 CON01370
C      CALL FOLINT(VERTCS,VT,FL,ILINE,IPTS,NI)                   CON01380
C      DO 250 LL=1,NI,2                                             CON01390
C      IB = (FL(LL) - SAMSTR) / SAMINC + 1                       CON01400
C      IE = (FL(LL+1) - SAMSTR) / SAMINC + 1                     CON01410
C      IF (MOD(SAMSTR,SAMINC) .NE. MOD(FL(LL),SAMINC)) IB = IB + 1 CON01420
C      IF (IP .GT. IE) GO TO 250                                   CON01430
C 115  DO 200 II=IR,IE                                             CON01440
C                                                                CON01450
C      DO 3 I=1,NC                                                  CON01460
C 3      JTEST(I) = 1                                              CON01470
C                                                                CON01480
C      FLOAT THE DATA SAMPLE, AND APPLY THE 'B'-MATRIX, IF AVAILABLE CON01490
C                                                                CON01500
C                                                                CON01510

```

FILE: CONTEX

```

C      IF ( BMFLAG ) GO TO 5
C      DO 4 I=1,NCHAN
C      IDUM = NPTS * ( I - 1 ) + 1
C      DATA(I) = IDATA(IDUM)
C      GO TO 15
C      5 DO 7 I=1,BMCOMB
C      SUM = 0.0
C      DO 6 K=1,NCHAN
C      INDUM = NPTS * ( K - 1 ) + 1
C      FDATA = IDATA(INDUM)
C      6 SUM = SUM + BMATR(I,K) * FDATA
C      7 DATA(I) = SUM
C      CALC. THE LIKLIHOOD VALUES (PROBABILITIES, IF YOU WILL)
C      15 TFMAX = -1.0E35
C      JI = 0
C      COMPUTE THE PDF FOR CLASS JJ
C      20 JJM1 = JJ - 1
C      LC = NPC * JJM1
C      LOCATION (-1) OF COV. MATRIX, CLASS JJ
C      IMN = JJM1 * NV
C      LOCATION (-1) OF MEAN VECTOR, CLASS JJ
C      KM = IMN + 1
C      S = DATA(1) - AVE(KM)
C      DM(1) = S
C      LC = LC + 1
C      TF = CON(JJ) + ( S * S )/COV(LC)
C      IF(KD1) GO TO 146
C      LOOP FOR COMPUTING THE KD-TH ELEMENT OF Y ( = L** - 1 * (X-M) ) , WHICH
C      IS STORED IN S
C      DO 145 KD=2,NV
C      KM = IMN + KD
C      S = DATA(KD) - AVE(KM)
C      JI = KD - 1
C      DO 140 LD = 1,JI
C      LC = LC + 1
C      140 S = S - COV(LC) * DM(LD)
C      DM(KD) = S
C      LC = LC + 1
C      COMPUTE THE KD-TH TERM IN :  $\frac{1}{2} * Y * D^{*-1} * Y$ 
C       $= \frac{1}{2} * (X-M) * K^{*-1} * (X-M)$ 
C      145 TF = TF + ( S * S )/COV(LC)
C      146 TF = -.5 * TF
C      TEST THIS SAMPLE PDF FOR CLASS JJ - IF GREATER THAN THE PDF FOR
C      CURRENT CLASS IC, SET IC = JJ. TEST THE CLASS-PAIR THRESHOLDS FOR
C      OTHER POSSIBLE CLASSES FOR THIS SAMPLE - IF THE PDF FOR CURRENT
C      CLASS IS EXCEEDED BY ANY CLASS-PAIR THRESHOLD, EVALUATE THE PDF
C      FOR THE OTHER CLASS OF THE CLASS-PAIR, AND REPEAT THE TEST FOR
C      MAX. PDF. IF ALL PDF'S FOR WHICH CLASS-PAIR THRESHOLDS HAVE
C      DICTATED TO BE TESTED HAVE BEEN EVALUATED, AND THE CURRENT PDF
C      FOR CLASS IC IS THE MAX. PDF OF ALL PDF'S EVALUATED,
C      CLASSIFY THE SAMPLE AS CLASS IC
C      IF( TF .LE. TFMAX ) GO TO 149
C      TFMAX = TF
C      IC = JJ
C      149 JTEST(JJ) = 0
C      JI = JI + 1
C      150 IF( JI .GT. NC ) GO TO 152
C      J = JORDER(JI)

```

CON01520
 CON01530
 CON01540
 CON01550
 CON01560
 CON01570
 CON01580
 CON01590
 CON01600
 CON01610
 CON01620
 CON01630
 CON01640
 CON01650
 CON01660
 CON01670
 CON01680
 CON01690
 CON01700
 CON01710
 CON01720
 CON01730
 CON01740
 CON01750
 CON01760
 CON01770
 CON01780
 CON01790
 CON01800
 CON01810
 CON01820
 CON01830
 CON01840
 CON01850
 CON01860
 CON01870
 CON01880
 CON01890
 CON01900
 CON01910
 CON01920
 CON01930
 CON01940
 CON01950
 CON01960
 CON01970
 CON01980
 CON01990
 CON02000
 CON02010
 CON02020
 CON02030
 CON02040
 CON02050
 CON02060
 CON02070
 CON02080
 CON02090
 CON02100
 CON02110
 CON02120
 CON02130
 CON02140
 CON02150
 CON02160
 CON02170
 CON02180
 CON02190
 CON02200
 CON02210
 CON02220
 CON02230
 CON02240
 CON02250
 CON02260
 CON02270
 CON02280
 CON02290

FILE: CONTEX

```

      IF ( JTEST(J) .EQ. 0 ) GO TO 150
      IF ( IC .GT. J ) GO TO 1501
      NTH = ( (J-1) * (J-2) )/2 + IC
      GO TO 1502
1501 NTH = ( (IC-1) * (IC-2) )/2 + J
1502 IF ( TFMAX .LT. TH(NTH) ) GO TO 151
      JTEST(J) = 0
      GO TO 150
151  JJ = J
      GO TO 20
152  JJ = IC
C
C   STORE THE CLASS NO. FOR THE SAMPLE, IN IR , AND THE VALUE OF THE
C   PDF FOR THE SAMPLE, IN VR .
C
      IR(JJ) = IC
      VR(JJ) = TFMAX
C
C
C   NCNT(IC) = NCNT(IC) + 1
C
200  CONTINUE
      IF ( LL .EQ. 1 ) IPT1 = IR(IB)
250  CONTINUE
C
C   SET PROBABLE CLASS FOR FIRST SAMPLE,NEXT SCAN LINE
C
C
C   ORDER THE CLASSES,ACCORDING TO FREQUENCY OF OCCURENCE ON CURRENT
C   SCAN LINE
C
      DO 201 I=1,NC
      JORDER(I) = I
      DO 210 I=1,NM1
      IPL1 = I + 1
      DO 205 K=IPL1,NC
      IF ( NCNT(I) .GT. NCNT(K) ) GO TO 205
      ITEMP = NCNT(I)
      NCNT(I) = NCNT(K)
      NCNT(K) = ITEMP
      ITEMP = JORDER(I)
      JORDER(I) = JORDER(K)
      JORDER(K) = ITEMP
205  CONTINUE
210  CONTINUE
C
C   RETURN
      END

```

CON02300
 CON02310
 CON02320
 CON02330
 CON02340
 CON02350
 CON02360
 CON02370
 CON02380
 CON02390
 CON02400
 CON02410
 CON02420
 CON02430
 CON02440
 CON02450
 CON02460
 CON02470
 CON02480
 CON02490
 CON02500
 CON02510
 CON02520
 CON02530
 CON02540
 CON02550
 CON02560
 CON02570
 CON02580
 CON02590
 CON02600
 CON02610
 CON02620
 CON02630
 CON02640
 CON02650
 CON02660
 CON02670
 CON02680
 CON02690
 CON02700
 CON02710
 CON02720
 CON02730
 CON02740
 CON02750
 CON02760
 CON02770
 CON02780

FILE: FALSY

```

SUBROUTINE FALSY(XL,XU,C,FXL,FXU,KC,XN,KT,T,K,KP1,S1,S2,U1,U2,RR) FAL00010
DIMENSION S1(K,K),S2(K,K),U1(K),U2(K),BB(K,KP1) FAL00020
RE=.1E-05*C FAL00030
IF(RF.LT..000001) RE=.000001 FAL00040
IF(KC.EQ.0) GO TO 7 FAL00050
FXL = G(XL,S1,S2,U1,U2,BB,KT,T,K,KP1) FAL00060
FXU = G(XU,S1,S2,U1,U2,BB,KT,T,K,KP1) FAL00070
7 R=(C-FXL)/(FXU-FXL) FAL00080
X=XL+R*(XU-XL) FAL00090
FX = G(X,S1,S2,U1,U2,BB,KT,T,K,KP1) FAL00100
FM=ABS(FX-C) FAL00110
I = 0 FAL00120
C GO TO 9 FAL00130
9 FXN = G(XN,S1,S2,U1,U2,BB,KT,T,K,KP1) FAL00140
I=I+1 FAL00150
E=ABS(FXN-C) FAL00160
C IF(E.GT.EM) GO TO 25 FAL00170
IF(E.LT.RE) RETURN FAL00180
EM=E FAL00190
IF(XN.LT.X) GO TO 12 FAL00200
XH=.5*(XU+XN) FAL00210
FXH = G(XH,S1,S2,U1,U2,BB,KT,T,K,KP1) FAL00220
IF(ABS(FXH-C).LT.RE) GO TO 30 FAL00230
IF(FXH.GT.C) GO TO 14 FAL00240
XL=XN FAL00250
FXL=FXN FAL00260
X=XH FAL00270
FX=FXH FAL00280
GO TO 15 FAL00290
14 XL=X FAL00300
FXL=FX FAL00310
X=XN FAL00320
FX=FXN FAL00330
XU=XH FAL00340
FXU=FXH FAL00350
GO TO 15 FAL00360
12 XH=.5*(XL+XN) FAL00370
FXH = G(XH,S1,S2,U1,U2,BB,KT,T,K,KP1) FAL00380
IF(ABS(FXH-C).LT.RE) GO TO 30 FAL00390
IF(FXH.GT.C) GO TO 13 FAL00400
XL=XH FAL00410
FXL=FXH FAL00420
XU=X FAL00430
FXU=FX FAL00440
X=XN FAL00450
FX=FXN FAL00460
GO TO 15 FAL00470
13 XU=XN FAL00480
FXU=FXN FAL00490
X=XH FAL00500
FX=FXH FAL00510
15 IF( I .EQ. 25) RETURN FAL00520
GO TO 9 FAL00530
25 XN=X FAL00540
RETURN FAL00550
30 XN=XH FAL00560
PFTUPH FAL00570
C FOLLOWING CODE FITS A QUADRATIC TO THE THREE POINTS FAL00580
C XL, X, XU WHICH IS AN APPROXIMATION OF THE FUNCTION G( H(X) ) FAL00590
C WITHIN THE DEFINED INTERVAL. A ROOT, XN, OF THE APPROXIMATING FAL00600
C QUADRATIC IS RETURNED TO BE USED AS A TRIAL SOLUTION OF FAL00610
C G( H(X) ) = C2 - C1 FAL00620
C FAL00630
R W1 = X - XL FAL00640
W2 = X*X - XL*XL FAL00650
W3=FX-FXL FAL00660
W4=XU-XL FAL00670
W5=XU*XU-XL*XL FAL00680
W6=FXU-FXL FAL00690
W7=C-FXL FAL00700
A=W1*W6-W3*W4 FAL00710
IF(ABS(A).LT.17.E-7) GO TO 44 FAL00720
R=W3*W5-W2*W6 FAL00730
F=-XL*XL*A-XL*W7*(W2*W4-W1*W5) FAL00740
D=R*R-4.*A*F FAL00750
IF(D.LT.0.) GO TO 44 FAL00760
D=SQRT(D) FAL00770
FAL00780
FAL00790

```

FILE: FALSY

```
      XN=(-B-D)/(2.0*A)
      IF(XN.GT.XU) XN=(-B-D)/(2.0*A)
      GO TO 9
44    XN=X
      GO TO 9
      END
```

```
FAL00800
FAL00810
FAL00820
FAL00830
FAL00840
FAL00850
```

1. **Introduction**
 2. **Background**
 3. **Methodology**
 4. **Results**
 5. **Conclusion**
 6. **References**
 7. **Appendix**
 8. **Index**
 9. **Table of Contents**
 10. **Summary**
 11. **Abstract**
 12. **Keywords**
 13. **Subject Headings**
 14. **Notes**
 15. **Footnotes**
 16. **References**
 17. **Appendix**
 18. **Index**
 19. **Table of Contents**
 20. **Summary**
 21. **Abstract**
 22. **Keywords**
 23. **Subject Headings**
 24. **Notes**
 25. **Footnotes**
 26. **References**
 27. **Appendix**
 28. **Index**
 29. **Table of Contents**
 30. **Summary**
 31. **Abstract**
 32. **Keywords**
 33. **Subject Headings**
 34. **Notes**
 35. **Footnotes**
 36. **References**
 37. **Appendix**
 38. **Index**
 39. **Table of Contents**
 40. **Summary**
 41. **Abstract**
 42. **Keywords**
 43. **Subject Headings**
 44. **Notes**
 45. **Footnotes**
 46. **References**
 47. **Appendix**
 48. **Index**
 49. **Table of Contents**
 50. **Summary**
 51. **Abstract**
 52. **Keywords**
 53. **Subject Headings**
 54. **Notes**
 55. **Footnotes**
 56. **References**
 57. **Appendix**
 58. **Index**
 59. **Table of Contents**
 60. **Summary**
 61. **Abstract**
 62. **Keywords**
 63. **Subject Headings**
 64. **Notes**
 65. **Footnotes**
 66. **References**
 67. **Appendix**
 68. **Index**
 69. **Table of Contents**
 70. **Summary**
 71. **Abstract**
 72. **Keywords**
 73. **Subject Headings**
 74. **Notes**
 75. **Footnotes**
 76. **References**
 77. **Appendix**
 78. **Index**
 79. **Table of Contents**
 80. **Summary**
 81. **Abstract**
 82. **Keywords**
 83. **Subject Headings**
 84. **Notes**
 85. **Footnotes**
 86. **References**
 87. **Appendix**
 88. **Index**
 89. **Table of Contents**
 90. **Summary**
 91. **Abstract**
 92. **Keywords**
 93. **Subject Headings**
 94. **Notes**
 95. **Footnotes**
 96. **References**
 97. **Appendix**
 98. **Index**
 99. **Table of Contents**
 100. **Summary**
 101. **Abstract**
 102. **Keywords**
 103. **Subject Headings**
 104. **Notes**
 105. **Footnotes**
 106. **References**
 107. **Appendix**
 108. **Index**
 109. **Table of Contents**
 110. **Summary**
 111. **Abstract**
 112. **Keywords**
 113. **Subject Headings**
 114. **Notes**
 115. **Footnotes**
 116. **References**
 117. **Appendix**
 118. **Index**
 119. **Table of Contents**
 120. **Summary**
 121. **Abstract**
 122. **Keywords**
 123. **Subject Headings**
 124. **Notes**
 125. **Footnotes**
 126. **References**
 127. **Appendix**
 128. **Index**
 129. **Table of Contents**
 130. **Summary**
 131. **Abstract**
 132. **Keywords**
 133. **Subject Headings**
 134. **Notes**
 135. **Footnotes**
 136. **References**
 137. **Appendix**
 138. **Index**
 139. **Table of Contents**
 140. **Summary**
 141. **Abstract**
 142. **Keywords**
 143. **Subject Headings**
 144. **Notes**
 145. **Footnotes**
 146. **References**
 147. **Appendix**
 148. **Index**
 149. **Table of Contents**
 150. **Summary**
 151. **Abstract**
 152. **Keywords**
 153. **Subject Headings**
 154. **Notes**
 155. **Footnotes**
 156. **References**
 157. **Appendix**
 158. **Index**
 159. **Table of Contents**
 160. **Summary**
 161. **Abstract**
 162. **Keywords**
 163. **Subject Headings**
 164. **Notes**
 165. **Footnotes**
 166. **References**
 167. **Appendix**
 168. **Index**
 169. **Table of Contents**
 170. **Summary**
 171. **Abstract**
 172. **Keywords**
 173. **Subject Headings**
 174. **Notes**
 175. **Footnotes**
 176. **References**
 177. **Appendix**
 178. **Index**
 179. **Table of Contents**
 180. **Summary**
 181. **Abstract**
 182. **Keywords**
 183. **Subject Headings**
 184. **Notes**
 185. **Footnotes**
 186. **References**
 187. **Appendix**
 188. **Index**
 189. **Table of Contents**
 190. **Summary**
 191. **Abstract**
 192. **Keywords**
 193. **Subject Headings**
 194. **Notes**
 195. **Footnotes**
 196. **References**
 197. **Appendix**
 198. **Index**
 199. **Table of Contents**
 200. **Summary**
 201. **Abstract**
 202. **Keywords**
 203. **Subject Headings**
 204. **Notes**
 205. **Footnotes**
 206. **References**
 207. **Appendix**
 208. **Index**
 209. **Table of Contents**
 210. **Summary**
 211. **Abstract**
 212. **Keywords**
 213. **Subject Headings**
 214. **Notes**
 215. **Footnotes**
 216. **References**
 217. **Appendix**
 218. **Index**
 219. **Table of Contents**
 220. **Summary**
 221. **Abstract**
 222. **Keywords**
 223. **Subject Headings**
 224. **Notes**
 225. **Footnotes**
 226. **References**
 227. **Appendix**
 228. **Index**
 229. **Table of Contents**
 230. **Summary**
 231. **Abstract**
 232. **Keywords**
 233. **Subject Headings**
 234. **Notes**
 235. **Footnotes**
 236. **References**
 237. **Appendix**
 238. **Index**
 239. **Table of Contents**
 240. **Summary**
 241. **Abstract**
 242. **Keywords**
 243. **Subject Headings**
 244. **Notes**
 245. **Footnotes**
 246. **References**
 247. **Appendix**
 248. **Index**
 249. **Table of Contents**
 250. **Summary**
 251. **Abstract**
 252. **Keywords**
 253. **Subject Headings**
 2

00010
00020
00030
00040
00050
00060
00070
00080
00090
00100
00110
00120
00130
00140
00150
00160
00170
00180
00190
00200
00210
00220
00230
00240
00250
00260
00270
00280
00290
00300
00310
00320
00330
00340
00350
00360
00370
00380
00390
00400
00410
00420
00430
00440
00450

FILE: GJR

```
SUBROUTINE GJR(A,NC,NR,N,MC,*,JC,V)
DIMENSION A(NR,NC),JC(1),V(2)
CALL OVERFL(IJK)
IW=V(1)
M=1
S=1.
L=N+(MC-N)*(IW/4)
KD=2-MOD(IW/2,2)
IF(KD.EQ.1)V(1)=0.
C JLS111069
KI=2-MOD(IW,2)
GO TO (5,20),KI
5 DO 10 I=1,N
10 JC(I)=I
20 DO 91 I=1,N
20 GO TO (22,21),KI
21 M=I
22 IF (I.EQ.N) GO TO 60
X=-1.
DO 30 J=1,N
IF (X.GT.ABS(A(J,I))) GO TO 30
X=ABS(A(J,I))
K=J
30 CONTINUE
IF(K.EQ.I) GO TO 60
S=-S
V(2)=-V(2)
C JLS111069
GO TO (35,40),KI
35 MU=JC(I)
JC(I)=JC(K)
JC(K)=MU
40 DO 50 J=M,L
X=A(I,J)
A(I,J)=A(K,J)
A(K,J)=X
50 IF (ABS(A(I,I)).GT.0.) GO TO 70
IF(KD.EQ.1)V(1)=0.
JC(I)=I-1
RETURN 1
70 GO TO (71,72),KD
71 IF(A(I,I).LT.0.) S=-S
V(1)=V(1)+ALOG(ABS(A(I,I)))
C JLS111069
72 X=A(I,I)
A(I,I)=1.
DO 80 J=M,L
A(I,J)=A(I,J)/X
CALL OVERFL(IFL)
IF(IFL.EQ.1) GO TO 150
80 CONTINUE
DO 91 K=1,N
IF (K.EQ.I) GO TO 91
X=A(K,I)
A(K,I)=0.
DO 90 J=M,L
A(K,J)=A(K,J)-X*A(I,J)
CALL OVERFL(IFL)
IF(IFL.EQ.1) GO TO 150
90 CONTINUE
91 CONTINUE
GO TO (95,140),KI
95 DO 130 J=1,N
IF (JC(J).EQ.J) GO TO 130
JJ=J+1
DO 100 I=JJ,N
IF (JC(I).EQ.J) GO TO 110
100 CONTINUE
110 JC(I)=JC(J)
DO 120 K=1,N
X=A(K,I)
A(K,I)=A(K,J)
A(K,J)=X
120 CONTINUE
130 JC(I)=N
140 RETURN
150 JC(I)=I-1
IF(KD.EQ.1)V(2)=S
```

GJR00010
GJR00020
GJR00030
GJR00040
GJR00050
GJR00060
GJR00070
GJR00080
GJR00090
GJR00100
GJR00110
GJR00120
GJR00130
GJR00140
GJR00150
GJR00160
GJR00170
GJR00180
GJR00190
GJR00200
GJR00210
GJR00220
GJR00230
GJR00240
GJR00250
GJR00260
GJR00270
GJR00280
GJR00290
GJR00300
GJR00310
GJR00320
GJR00330
GJR00340
GJR00350
GJR00360
GJR00370
GJR00380
GJR00390
GJR00400
GJR00410
GJR00420
GJR00430
GJR00440
GJR00450
GJR00460
GJR00470
GJR00480
GJR00490
GJR00500
GJR00510
GJR00520
GJR00530
GJR00540
GJR00550
GJR00560
GJR00570
GJR00580
GJR00590
GJR00600
GJR00610
GJR00620
GJR00630
GJR00640
GJR00650
GJR00660
GJR00670
GJR00680
GJR00690
GJR00700
GJR00710
GJR00720
GJR00730
GJR00740
GJR00750
GJR00760
GJR00770
GJR00780
GJR00790

ORIGINAL PAGE IS
OF POOR QUALITY

FILE: GJR

RETURN 1
END

GJR00800
GJR00810

~~11-24~~

160

FILE: MAPHDG

```

SUBROUTINE MAPHDG(NOCAT,CLSSYM,CATNAM,KATNO,CLSMTX,SURNO,SURDES)
THIS ROUTINE PRINTS THE HEADER INFORMATION FOR THE CLASSIFICATION
MAP IN CLASSIFY AND DISPLAY

      NOCAT -- NO. OF CATEGORIES
      CLSSYM -- SYMBOLS FOR CATEGORIES OR SURCLASSES
      CATNAM -- CATEGORY NAMES
      KATNO -- CATEGORY EACH CLASS WAS ASSIGNED TO
      CLSMTX -- CLASS NAMES
      SURNO -- NO. OF SURCLASSES IN EACH CLASS
      SURDES -- SURCLASS NAMES
      CLSVC2 -- CLASS EACH SUBCLASS WAS ASSIGNED TO (IN COMMON
      BLOCK INFORM)

      IMPLICIT INTEGER (A-Z)

      INCLUDE COMCHK1.LIST
      COMMON/INFORM/NOCLS2,NOSUR2,NOFET2,VARSZ2,TOTVT2,NOFLD2,
      *      AVAR2,CVAR2,CLSID2,SURNO2,SURDS2,FLDSV2,VERTX2,
      *      FETVC2(30),SUHVC2(75),SUBPTH(75),CLSVC2(60),
      *      KPPPTS(60),NOGRP,GHPNAM(60),GRPDEX(61),
      *      GRPCHK(61),GROUPS(124)

      LOGICAL ISWTH
      DIMENSION CLSSYM(1),CATNAM(1),KATNO(1),CLSMTX(1),SUBNO(1),
      *      SURDES(1)

      PRINTS CATEGORY CLASSIFIER INFORMATION

      IF (NOCAT .EQ. 0) GO TO 82
      WRITE(6,200)
200  FORMAT(// T42,'MAP OF CATEGORY CLASSIFIER CLASSIFICATION RESULTS',
      *      // T32,'CATEGORY',T62,'CLASS',T93,'SURCLASS',T31,
      *      'NO.',T37,'NAME',T60,'NO.',T66,'NAME',
      *      T94,'NO.',T94,'NAME',T101,'SYMBOL')

      DO 68 I=1,NOCAT
      WRITE(6,210) I,CATNAM(I)
210  FORMAT(/T31,I2,T37,A4)
      ISWTH = .TRUE.
      DO 63 J=1,NOCLS2
      IF (KATNO(J) .EQ. I) GO TO 64
      GO TO 63
64  IF (ISWTH) GO TO 65
      WRITE(6,220) J,CLSMTX(J)
220  FORMAT(/T40,I2,T66,A4)
      ISWTH = .TRUE.
      GO TO 66
65  WRITE(6,230) J,CLSMTX(J)
230  FORMAT(1H+,T40,I2,T66,A4)
66  DO 67 K=J,NOSUR2
      IF (CLSVC2(K) .EQ. J) GO TO 70
      GO TO 67
70  NSURCL = SURNO(J)
      KK = 0
      DO 75 L=1,NSURCL
      KK = K + L - 1
      IF (ISWTH) GO TO 72
      WRITE(6,250) KK,SURDES(KK),CLSSYM(KK)
250  FORMAT(1H+,I2,T94,A4,T103,A1)
      GO TO 75
72  WRITE(6,240) KK,SURDES(KK),CLSSYM(KK)
240  FORMAT(1H+,T94,I2,T94,A4,T103,A1)
      ISWTH = .FALSE.
75  CONTINUE
      GO TO 63
67  CONTINUE
63  CONTINUE
68  RETURN

      PRINTS STANDARD CLASSIFIER INFORMATION

```

FILE: MAPHOG

| | | |
|-----|---|----------|
| A2 | CONTINUE | MAP00760 |
| C | | MAP00770 |
| | WRITE(4,260) | MAP00780 |
| 260 | FORMAT(// T42, 'MAP OF STANDARD CLASSIFIER CLASSIFICATION RESULT | MAP00790 |
| | *S1 ///T45, 'CLASS', T77, 'SUBCLASS' / T42, 'NO.', T50, 'NAME', T72, 'NO.', | MAP00800 |
| | *T78, 'NAME', T85, 'SYMBOL') | MAP00810 |
| C | | MAP00820 |
| | CLSNUM = 1 | MAP00830 |
| | ISWTH = .TRUE. | MAP00840 |
| | DO A9 I=1, NOSUB2 | MAP00850 |
| | IF (CLSNUM.EQ. CLSVC(I)) GO TO A5 | MAP00860 |
| | CLSNUM = CLSNUM + 1 | MAP00870 |
| | GO TO A7 | MAP00880 |
| A5 | IF (ISWTH) GO TO A7 | MAP00890 |
| | WRITE(4,270) I, SURDES(I), CLSSYM(I) | MAP00900 |
| 270 | FORMAT(T72, I2, T78, A4, T87, A1) | MAP00910 |
| | GO TO A9 | MAP00920 |
| A7 | WRITE(4,240) CLSNUM, CLSMTX(CLSNUM), I, SURDES(I), CLSSYM(I) | MAP00930 |
| 240 | FORMAT(//T42, I2, T50, A4, T72, I2, T78, A4, T87, A1) | MAP00940 |
| | ISWTH = .FALSE. | MAP00950 |
| A9 | CONTINUE | MAP00960 |
| | RETURN | MAP00970 |
| | END | MAP00980 |

FILE: MCMLSK

SUBROUTINE MCMLSK(KK,NV,DUM,DET)

THIS ROUTINE COMPUTES THE MODIFIED CHOLESKY DECOMPOSITION OF THE COVARIANCE MATRIX. THE DECOMPOSITIONS OVERLAY THE ELEMENTS OF THE COVARIANCE MATRIX.

KK = L D L*

KK = COVARIANCE MATRIX STORED IN SYMMETRIC STORAGE

NV = NO. OF CHANNELS

DUM = A WORK AREA OF SIZE NV-1

DET = THE DETERMINANT OF THE COVARIANCE MATRIX

REAL KK
LOGICAL JE1
DIMENSION KK(1), DUM(1)

DOUBLE PRECISION TF, R, R1, DUM, T1

JE1 = .TRUE.

J1 = 0

JD = 0

DET = 1.0

LOOP OVER ALL CHANNELS

DO 10 J=1,NV

KL = J-1

L = J+1

JD = J+1

J1 = J1 + J

TF = KK(J1)

IF (JE1) GO TO 12

K1 = 0

COMPUTE THE DIAGONAL ELEMENTS OF D AND STORE IN KK

TEMPORARILY STORE THE PRODUCT KK(I,I)*KK(J,I) IN DUM(I)

DO 15 I=1,KL

R = KK(JD + I)

K1 = K1 + I

R1 = KK(K1) * R

TF = TF - R1 * R

DUM(I) = K1

CONTINUE

KK(J1) = TF

CONTINUE

DET = DET * TF

IF (L .GT. NV) GO TO 10

IRD = J1 - L + 1

COMPUTE THE R, J-TH ELEMENT OF L, USING T1

DO 20 IRD = L,NV

IRD = IRD + IR - 1

T1 = KK(IRD + J)

IF (JE1) GO TO 16

DO 25 I=1,KL

T1 = T1 - DUM(I) * KK(IRD + I)

CONTINUE

KK(IRD + J) = T1/TF

CONTINUE

JE1 = .FALSE.

CONTINUE

KK CONTAINS, IN SYMMETRIC STORAGE, THE MODIFIED CHOLESKY FACTORIZATION OF THE INPUT MATRIX. THE LOWER TRIANGULAR MATRIX, L, OCCUPIES THE OFF-DIAGONAL ELEMENTS OF KK, AND THE DIAGONAL MATRIX, D, IS STORED IN THE DIAGONAL ELEMENTS IN KK.

RETURN

END

[illegible]

[illegible]

~~11-29~~
165

FILE: REDIF2

```
C
240 M = NXTCHR(CARD2,COL)
    IF ( M.EQ. BLANK ) GO TO 225
    COL = COL - 1
    NOSUB2 = NUMBER(CARD2,COL,SURVC2,NOSUB2)
    CALL ORDER(SURVC2,NOSUB2)

C
    GO TO 225

C
    CHANNELS
    -----
250 IF ( RMFLG.GT. 0) GO TO 225
    M = NXTCHR( CARD2, COL )
    IF ( M.EQ. BLANK ) GO TO 225
    J = FIND12(CARD2,COL,EQUEVC)
    IF (J.EQ. -1) GO TO 251
    IF (M.EQ. SPCD) GO TO 253
    IF (M.EQ. D4CD) GO TO 255
251 WRITE(6,252)
252 FORMAT(' ERROR ON CHANNELS CARD*')
    GO TO 225
253 NF = NUMBER(CARD2,COL,FETVC2,NF)
    NOFET2 = NF
    CALL ORDER(FETVC2,NOFET2)
    COL = COL - 1
    GO TO 250
255 NOCHAN = NUMBER(CARD2,COL,CHNVEC,NOCHAN)
    CALL ORDER(CHNVEC,NOCHAN)
    COL = COL - 1
    GO TO 250

C
    DATE
    -----
260 M = NXTCHR(CARD2,COL)
    IF ( M.EQ. BLANK ) GO TO 225
    READ (30,9999) DATE
    REWIND 30
9999 FORMAT(10X,11A4)
    GO TO 225

C
    COMMENT
    -----
500 CONTINUE
    READ (30,9999) COMENT
    REWIND 30
    GO TO 225

C
    HED1
    -----
510 CONTINUE
    READ (30,9999) HED1
    REWIND 30
    GO TO 225

C
    HED2
    -----
520 CONTINUE
    READ (30,9999) HED2
    REWIND 30
    GO TO 225

C
    OPTION CARD
    -----
530 M = NXTCHR(CARD2,COL)
    IF ( M.EQ. BLANK ) GO TO 225
    IF ( M.EQ. SRCD ) GO TO 1000
    STATKY = 1
    M = FIND12( CARD2,COL,COMVEC)
    IF ( M.EQ. 0) GO TO 225
    GO TO 530
MODULE
    -----
```

RED01590
RED01600
RED01610
RED01620
RED01630
RED01640
RED01650
RED01660
RED01670
RED01680
RED01690
RED01700
RED01710
RED01720
RED01730
RED01740
RED01750
RED01760
RED01770
RED01780
RED01790
RED01800
RED01810
RED01820
RED01830
RED01840
RED01850
RED01860
RED01870
RED01880
RED01890
RED01900
RED01910
RED01920
RED01930
RED01940
RED01950
RED01960
RED01970
RED01980
RED01990
RED02000
RED02010
RED02020
RED02030
RED02040
RED02050
RED02060
RED02070
RED02080
RED02090
RED02100
RED02110
RED02120
RED02130
RED02140
RED02150
RED02160
RED02170
RED02180
RED02190
RED02200
RED02210
RED02220
RED02230
RED02240
RED02250
RED02260
RED02270
RED02280
RED02290
RED02300
RED02310
RED02320
RED02330
RED02340
RED02350
RED02360
RED02370

FILE: REDIF2

```
C
540 CONTINUE
    CALL CRDSTA (ARRAY, TOP)
    DATSW = 1
    GOTO 225
C
C
C    GROUPS
C    -----
560 CONTINUE
    I = GRPSCN (CARD2, SYMMAX, GRPTR)
    GOTO 225
C
C    B-MATRIX INPUT
C
580 M = FIND12 (CARD2, COL, BMCOF)
    IF (M .EQ. 2) GO TO 581
    B-MATRIX INPUT FROM CARDS
    IF (M .EQ. 3) GO TO 582
    B-MATRIX INPUT FROM FILE
C
    WRITE (6, 5801) CODE, CARD2
5801 FORMAT (//// 5X, '***' CLSFY/REDIF2 --- B-MATRIX INPUT STIPULATED
1 BY CONTROL CARD ... // 5X, 2H 'A4.6X.62A1.2H ' // 5X,
2 '***' UNABLE TO INTERPRET TYPE OF B-MATRIX INPUT --- PROGRAM EXEC
3UTION TERMINATED FROM REDIF2 '***' /1H1)
C
    CALL EXIT
C
C
C    B-MATRIX INPUT ON CARDS
C
581 CALL RMFIL (RMATRIX, BMCOMB, BMFEAT, FETVC2, 1)
    NOFFT2 = BMFEAT
    RMKEY = 1
    RMFLG = (BMCOMB * (BMCOMB + 1)) / 2
    IF (BMFEAT .LE. 0 .OR. BMCOMB .LE. 0) GO TO 586
    NRM = BMCOMB * BMFEAT
    IF (NRM .GT. 450) GO TO 586
    RMFLG = (BMCOMB * (BMCOMB + 1)) / 2
C
    END FILE RMFILE
    RMKEY = 1
    GO TO 225
C
C
C    B-MATRIX INPUT FROM B-MATRIX FILE
C
582 CALL RMFIL (RMATRIX, BMCOMB, BMFEAT, FETVC2, 2)
    NOFFT2 = BMFEAT
C
    IF (BMFEAT .LE. 0 .OR. BMCOMB .LE. 0) GO TO 587
    NRM = BMCOMB * BMFEAT
    IF (NRM .GT. 450) GO TO 587
C
    RMFLG = (BMCOMB * (BMCOMB + 1)) / 2
    GO TO 225
C
C
C    ERROR RETURNS --- B-MATRIX INPUT
C
586 WRITE (6, 5861) CODE, CARD2
5861 FORMAT (//// 5X, '***' CLSFY/REDIF2 --- BAD CARD INPUT DETECTED ON
1 ATTEMPT TO READ B-MATRIX INFORMATION AS DIRECTED BY THE CONTROL CAR
2D ... // 5X, 2H 'A4.6X.62A1.2H ' // 5X, '***' TERMINATING PROGRAM EXEC
3UTION FROM REDIF2 '***' /1H1)
C
    CALL EXIT
C
587 WRITE (6, 5871) BMCOMB, BMFEAT, (FETVC2(I), I=1, RMFEAT)
5871 FORMAT (//// 5X, '***' CLSFY/REDIF2 --- B-MATRIX INPUT FROM RMFILE -
2 // 2X, 'BAD INPUT VALUES DETECTED: NO. COMBINATIONS (BMCOMB) = ', I5,
3 1X, ' NO. CHANNELS (BMFEAT) = ', I5, 1X, ' CHANNEL VECTOR (BMVEC) = ',
4 ... // ( 5X, 30I4 ) )
    WRITE (6, 5872)
```

[illegible]


```

IF (STAFIL - LT. 0) STAFIL = 0
COL = COL - 1
GO TO 620

-----

630 CONTINUE
    ERPKEY = 1
    J = NXTCHR(CARD2,COL)
631 IF (J.EQ.BLANK) GO TO 650
    IF (J.NE.URCD) GO TO 635
    J = FIND12(CARD2,COL,EQUVEC)
    IF (J.NE.2) GO TO 650
    ISTART = 0
    J = NUMBER(CARD2,COL,MAPTAP,ISTART)
    J = FIND12(CARD2,COL,EQUVEC)
    IF (J.NE.2) GO TO 650
    ISTART = 0
    J = NUMBER(CARD2,COL,ERPKEY,ISTART)
    GO TO 225
635 IF (J.NE.URCD) GO TO 640
    J = FIND12(CARD2,COL,SLASH)
    IF (J.NE.2) GO TO 650
    J = NXTCHR(CARD2,COL)
    IF (J.EQ.FRCD) GO TO 645
    IF (J.EQ.URCD) GO TO 631
    GO TO 650
640 IF (J.NE.FRCD) GO TO 650
645 J = FIND12(CARD2,COL,EQUVEC)
    IF (J.NE.2) GO TO 650
    ISTART = 0
    J = NUMBER(CARD2,COL,ERPKEY,ISTART)
    J = FIND12(CARD2,COL,EQUVEC)
    IF (J.NE.2) GO TO 650
    ISTART = 0
    J = NUMBER(CARD2,COL,MAPTAP,ISTART)
    GO TO 225
650 WRITE (6,655)
655 FORMAT(' ERROR ON MAPTAP CONTROL CARD')
    GO TO 225
270 CONTINUE
290 CONTINUE
    IF (NOCAT.EQ.1) WRITE(6,615)
    IF (NOCAT.EQ.1) CALL CMERR
615 FORMAT('/// 5X, *MUST HAVE AT LEAST TWO CATEGORIES')
    RETURN

ERROR ROUTINES
-----
1000 WRITE (6,10002) CODE, CARD2
10002 FORMAT('/// 5X, ***** CLSFY/REDIF2 --- BAD PROCESSOR CONTROL CARRED04500
IN .....//5X,2H' ,A4,6X,62A1,2H'//5X,***** TERMINATING PROGRAM EXPRED04510
2ECUTION FROM REDIF2 *****//1H1)
GO TO 225

END

```

FILE: RELERR

```

FUNCTION RELERR(COVMTX, COV, NOFET2, VARSZ2)
  INTEGER VARSZ2
  COMMON/SCRACH/SCR1(2000), SCR2(10500)
  DIMENSION D(30), COV(VARSZ2), COVMTX(VARSZ2)
  REAL LDL(465)
  EQUIVALENCE (ENORMO, SCR2(1)), (ENORMD, SCR2(2)) ,
C 1 ( D(1), SCR2(3)), ( LDL(1), SCR2(33)),
C 2 ( SUM, SCR2(963)), ( II, SCR2(964)), ( I, SCR2(965)),
C 3 ( J, SCR2(966)), ( L, SCR2(967)), ( JJ, SCR2(968)),
C 4 ( KK, SCR2(969)), ( JK, SCR2(970)), ( KP, SCR2(971)),
C 5 ( III, SCR2(972)), ( IJ, SCR2(973)), ( JP, SCR2(974)),
C 6 ( IP, SCR2(975))
C
C  COMPUTE THE EUCLIDEAN NORM OF THE COVARIANCE MATRIX, BEFORE
C  CHOLESKY FACTORIZATION
C
  ENORMO = 0.0
  DO 181 I=1, VARSZ2
C 181 ENORMO = ENORMO + 2.0 * COV(I) * COV(I)
C
  II = 0
  DO 182 I=1, NOFET2
  II = II + I
  ENORMO = ENORMO - ( COV(II) * COV(II) )
  D(I) = COVMTX(II)
C 182 COVMTX(II) = 1.0
C
  ENORMO = SQRT(ENORMO)
C
  IJ = 0
  DO 187 I=1, NOFET2
  IK = I
  II = ( IK * (IK-1) )/2
  DO 186 J=1, IK
  JK = J
  SUM = 0.0
  JJ = ( JK * (JK-1) )/2
  DO 185 KP=1, JK
  KK = KP
  JP = JJ + KP
  IP = II + KP
C 185 SUM = SUM + ( COVMTX(JP) * COVMTX(IP) * D(KP) )
C 186 LDL(IJ) = SUM
C 187 LDL(IJ) = L * D * L *
C
  CONTINUE
C
  II = 0
  DO 188 L=1, NOFET2
  II = II + L
C 188 COVMTX(II) = D(L)
  ENORMD = 0.0
  III = 0
  II = 0
  DO 190 I=1, NOFET2
  II = II + I
  DO 189 J=1, I
  III = III + 1
  SUM = COV(III) - LDL(III)
C 189 ENORMD = ENORMD + 2.0 * SUM * SUM
  SUM = COV(II) - LDL(II)
C 190 ENORMD = ENORMD - ( SUM * SUM )
C
  IF ( ENORMD .LE. 1.0E-8) GO TO 191
  ENORMD = SQRT(ENORMD)
C
C 191 RELERR = ENORMD/ENORMO
  RETURN
  END

```


[illegible]

```

DATA ENDRCD/'SEND'//
DATA CLSSY/'1','2','3','4','5','6','7','8','9','A','B','C','D',
          'E','F','G','H','I','J','K','L','M','N','O','P','Q',
          'R','S','T','U','V','W','X','Y','Z','_','-','/','.',
          ':','>','<',' ','(',')','{','}','[',''],'\','|','^','`','~','!','@','#','$','%','&','*','+','=','>',
          '<','>'//
DO 5 I=1,60
5 CLSSYM(I)=CLSSY(I)
-----
DIMENSION BMATRIX(RMCOMB,BMFEAT)
-----
-----
-----
NOCLSS=0
FILEOP = 0
GO READ SUPERVISOR INFORMATION
-----
CALL REDIF2(ARRAY,TOP,APRIOR,KATNO,BMATRIX,PRIORI)
GET TAPE READY
310 REWIND MAPTAP
      NFILE = ERPKEY
      IF (NFILE.LE.0) NFILE = 1
      ISKIP = NFILE - 1
      IF (ISKIP.LE.0) GO TO 400
      CALL FSBSFL(MAPTAP,ISKIP,ISTAT)
      IF (ISTAT.NE.0) CALL EXIT
400 CONTINUE
-----
CALL ROUTINE TO REDUCE THE ARRAYS
-----
CALL REDSAV(ARRAY,TOP,RMFLG)
TEST FOR CATEGORY 'FILE' INPUT -IF SO FORM CATEGORY/CLASS/SUBCLASS
ASSOCIATION USING CLASS/SUBCLASS INFORMATION.
IDUM = -7654321
IF (NOCAT.NE.IDUM) GO TO 406
FILEOP = 1
NOCAT = NOCLS2
DO 405 I=1,NOCLS2
P=CLSID2 + I -1
CATNAM(I) = ARRAY(P)
NOCTCL(I) = 1
KCLSNA(I) = ARRAY(P)
405 CONTINUE
406 CONTINUE
-----
DEFAULT -- SAME CHANNELS AS SELECTED FROM STAT FILE
IF (NOCHAN .NE. 0) GO TO 335
DO 330 I=1,NOFET2
330 CHNVEC(I) = FETVC2(I)
NOCHAN = NOFET2
335 CONTINUE
-----
COMPUTE BASE ADDRESSES FOR SCRACH ARRAY
THIJ1 - BASE ADDRESS FOR TABLE COMPUTED IN THRESH CONTAINING
CLASS-PAIR THRESHOLDS

```

ORIGINAL PAGE IS
OF POOR QUALITY

FILE SETUP2

```

C      IDATA1- BASE ADDRESS FOR DATA PASSED BACK FROM TAPERD
C      THIJ1 = 1
C      IDATA1 = THIJ1 + (NOSUR2-1)*(NOSUB2-2)/2 + NOSUB2
C      IF (NOCAT.GT. 0) IDATA1 = 1
C
C      STORE A BLANK IN DEFAULT SYMBOLS. THIS WILL BE USED IN
C      PRINTING THE MAP CLASSIFICATION FOR THE UNCLASSIFIED PIXEL
C
C      CLSSYM(NOSUB2 + 1) = BLANK
C      IF (NOCAT.LE. 0) GO TO 465
C
C      ALL CLASSES MUST BE ASSIGNED TO A CATEGORY
C
C      DO 410 II=1,NOCAT
C      NOCLSS = NOCTCL(II) + NOCLSS
C      IF (NOCLSS.EQ. NOCLS2) GO TO 415
C      WRITE(6,450) (ARRAY(CLSID2+I-1),I=1,NOCLS2)
C      WRITE(6,460) (KCLSNA(I),I=1,NOCLSS)
C      CALL CMERR
C
C      SET UP KATNO ARRAY TO CONTAIN THE CATEGORY EACH CLASS
C      BELONGS TO
C
C      415 CONTINUE
C      NOCLAS = 0
C      DO 435 L=1,NOCAT
C      NOCLS1 = NOCLAS + 1
C      NOCLAS = NOCTCL(L) + NOCLS1 - 1
C      DO 425 J=NOCLS1,NOCLAS
C      CLSNAM = KCLSNA(J)
C      DO 420 K=1,NOCLS2
C      IF (CLSNAM.EQ. ARRAY(CLSID2-1+K)) GO TO 430
C      420 CONTINUE
C      440 WRITE(6,450) (ARRAY(CLSID2+I-1),I=1,NOCLS2)
C      450 FORMAT(// 'AN ERROR HAS OCCURRED IN GROUPING CLASSES INTO CATEGORIES'
C      * 'YES.CHECK THE FOLLOWING : '// 5X,'1. NOT ALL OF THE CLASSES HAVE BEEN
C      * 'EN ASSIGNED TO A CATEGORY. '// 5X,'2. A CLASS NAME ON THE CATEGORY
C      * 'CARD HAS BEEN MISPELLED. '// 10X,'CLASS NAMES FROM SAVTAP FILE ARSE
C      * 'E : ' / 5X, (10(A6,2X)))
C      WRITE(6,460) (KCLSNA(I),I=1,NOCLSS)
C      460 FORMAT( / 10X, 'CLASS NAMES FROM CATEGORY CARDS ARE : '// 5X,
C      * (10(A4,4X)))
C      CALL CMERR
C      430 KATNO(K) = L
C      425 CONTINUE
C      435 CONTINUE
C
C      SET UP SUBCAT ARRAY TO CONTAIN THE CATEGORY EACH SUBCLASS
C      BELONGS TO
C
C      DO 437 II = 1,NOSUR2
C      CLSNUM = CLSVC2(II)
C      437 SURCAT(II) = KATNO(CLSNUM)
C      465 CONTINUE
C
C      PRINT OUT THE SUPERVISOR INFORMATION
C      -----
C
C      WRITE(6,HEAD)
C      WRITE(6,5012)
C      IF (STATKY.EQ. 1) WRITE(6,5014)
C      IF (NOCAT.GT. 0) WRITE(6,5016)
C      IF (NOCAT.LE. 0) WRITE(6,5020)
C      IF (FILEOP.EQ.1) WRITE(6,5022)
C      5012 FORMAT(12,'THE FOLLOWING OPTIONS HAVE BEEN SELECTED'//)
C      5014 FORMAT(15,'PRINT MULTISPECTRAL STATISTICS.//)
C      5016 FORMAT(15,'CATEGORY CLASSIFIER OPTION HAS BEEN SELECTED.//)
C      5020 FORMAT(15,'STANDARD CLASSIFIER OPTION HAS BEEN SELECTED.//)
C      5022 FORMAT(17,'ALSO CLASSES FROM STATFILE WILL BE CONSIDERED THE CATEG
C      * 'ORIES FOR CLASSIFICATION')
C      500 CONTINUE
C      502 FORMAT( // 1X,'SUPERVISOR INFORMATION : '//15,'FILE NUMBER ....',
C      * 110/ 15,'NO. OF FIELDS.....',

```

SET01530
SET01540
SET01550
SET01560
SET01570
SET01580
SET01590
SET01600
SET01610
SET01620
SET01630
SET01640
SET01650
SET01660
SET01670
SET01680
SET01690
SET01700
SET01710
SET01720
SET01730
SET01740
SET01750
SET01760
SET01770
SET01780
SET01790
SET01800
SET01810
SET01820
SET01830
SET01840
SET01850
SET01860
SET01870
SET01880
SET01890
SET01900
SET01910
SET01920
SET01930
SET01940
SET01950
SET01960
SET01970
SET01980
SET01990
SET02000
SET02010
SET02020
SET02030
SET02040
SET02050
SET02060
SET02070
SET02080
SET02090
SET02100
SET02110
SET02120
SET02130
SET02140
SET02150
SET02160
SET02170
SET02180
SET02190
SET02200
SET02210
SET02220
SET02230
SET02240
SET02250
SET02260
SET02270
SET02280

FILE SETUP2

```

2 16/T5,'NO. OF CLASSES.....',14/T5,'NO. OF SUBCLASSES.....'
3 14/T5,'NO. OF CHANNELS.....',14/77777)
C
C
C   IF (APRFLG .LE. 0 ) GO TO 6045
C   NORMALIZE APRIORI VALUES THAT WERE INPUT BY USER
C   NORM = 0.0
C   DO 600 I=1,APRFLG
600  NORM = NORM + ABS(PRIORI(I))
C   IF ( NORM .LT. .999999 .OR. NORM .GT. 1.000001) WRITE(6,6041)
6041  * VALUES WERE NORMALIZED **
C   DO 603 I=1,APRFLG
603  PRIORI(I) = ABS(PRIORI(I))/NORM
C   IF (APRFLG .EQ. NOSUB2) GO TO 610
C   IF (APRFLG .EQ. NOCLS2) GO TO 620
C   IF (APRFLG .EQ. NOCAT) GO TO 630
C
C   WRITE(6,6010)
6010  FORMAT(15,'** ERROR IN A PRIORI CONTROL CARD. USER INPUT VALUES IG
      *NORED. **')
      GO TO 6045
C
C   APRIORI VLAUES INPUT BY SUBCLASSES
C
610  DO 615 I=1,NOSUB2
615  APRIOR(I) = PRIORI(I)
      WRITE(6,6015)
6015  FORMAT(15,'APRIORI VALUES INPUT BY SUBCLASSES  ')
      GO TO 6070
C
C   APRIORI VALUES INPUT BY CLASSES
C
620  JJ = 0
      DO 625 I=1,NOCLS2
      NOCL = ARRAY(SUBNO2-1+I)
      APRI = PRIORI(I) / FLOAT(NOCL)
      DO 625 J=1,NOCL
      JJ = JJ + 1
625  APRIOR(JJ) = APRI
      WRITE(6,6025)
6025  FORMAT(15,'APRIORI VALUES INPUT BY CLASSES.  APRIOR(I) = CLASS(J
      *) APRIOR / (NO. SUBCLASSES IN CLASS(J)) ')
      GO TO 6070
C
C   APRIORI VALUES INPUT BY CATEGORIES
C
630  CLSNM = 0
      DO 660 I=1,NOCAT
      APRI = PRIORI(I)
      II = NOCTCL(I)
      M = 1
      NOSBCL = 0
      DO 645 K=1,II
      DO 640 KK=1,NOCLS2
      IF (KCLSNA(K+CLSNM) .EQ. ARRAY(CLSID2-1+KK)) GO TO 645
640  CONTINUE
645  NOSBCL = ARRAY(SUBNO2-1+KK) + NOSBCL
      APRIO = APRI / FLOAT(NOSBCL)
      DO 650 KKK=1,NOSUB2
      IF ( I .EQ. SURCAT(KKK)) APRIOR(KKK) = APRIO
      IF ( I .EQ. SURCAT(KKK)) M = M + 1
      IF (M .GT. NOSBCL) GO TO 655
650  CONTINUE
655  CLSNM = CLSNM + II
660  CONTINUE
      WRITE(6,6060)
6060  FORMAT(15,'APRIORI VALUES INPUT BY CATEGORY.  APRIOR(I) = CATEGO
      *RY(J) APRIOR / (NO. SURCLASSES IN CATEGORY(J)) ')
      GO TO 6070
C
C   6045 IF (APRFLG.EQ.-777777)GO TO 760
C   FLAG FOR SAVTAP COMPUTATION
C   IF (NOCAT.LE. 0) GO TO 605
C

```

```

SET022900
SET023000
SET023100
SET023200
SET023300
SET023400
SET023500
SET023600
SET023700
SET023800
SET023900
SET024000
SET024100
SET024200
SET024300
SET024400
SET024500
SET024600
SET024700
SET024800
SET024900
SET025000
SET025100
SET025200
SET025300
SET025400
SET025500
SET025600
SET025700
SET025800
SET025900
SET026000
SET026100
SET026200
SET026300
SET026400
SET026500
SET026600
SET026700
SET026800
SET026900
SET027000
SET027100
SET027200
SET027300
SET027400
SET027500
SET027600
SET027700
SET027800
SET027900
SET028000
SET028100
SET028200
SET028300
SET028400
SET028500
SET028600
SET028700
SET028800
SET028900
SET029000
SET029100
SET029200
SET029300
SET029400
SET029500
SET029600
SET029700
SET029800
SET029900
SET030000
SET030100
SET030200
SET030300
SET030400

```

ORIGINAL PAGE IS
OF POOR QUALITY

FILE SETUP2

```
C      COMPUTE DEFAULT APRIORI VALUE FOR CATEGORY CLASSIFIER
C      CLSNM = 0
      APR1 = 1.0 / FLOAT(NOCAT)
      DO 750 I=1,NOCAT
        II = NOCTCL(I)
        M = 1
        NOSBCL = 0
        DO 715 K=1,II
          DO 705 KK=1,NOCLS2
            IF (KCLSNA(K*CLSNM) .EQ. ARRAY(CLSID2-1*KK)) GO TO 715
          705 CONTINUE
          715 NOSBCL = ARRAY(SURN02-1*KK) + NOSBCL
          APR10 = APR1 * 1.0 / FLOAT(NOSBCL)
          DO 720 KKK=1,NOSUR2
            IF (I .EQ. SURCAT(KKK)) APRIOR(KKK) = APR10
            IF (I .EQ. SURCAT(KKK)) M = M + 1
            IF (M .GT. NOSBCL) GO TO 740
          720 CONTINUE
          740 CLSNM = CLSNM + II
          750 CONTINUE
        GO TO 6053

C      COMPUTE APRIORI VALUES FROM STATFILE
C      760 TKEPTS = 0
      DO 765 I=1,NOSUR2
        765 TKEPTS = TKEPTS + KEPPTS(I)
      C      TOTAL ALL SUBCLASS PIXELS
      DO 770 I=1,NOSUR2
        APRIOR(I) = FLOAT(KEPPTS(I))/FLOAT(TKEPTS)
      770 CONTINUE
      IPAT = STAFIL * 1
      WRITE(6,775) IPAT
      775 FORMAT(15,'APRIORI VALUES FROM STATFILE',I3,' APRIORI= NO. PI
        *XELS IN SURCLASS/TOTAL NO. PIXELS IN ALL SUBCLASSES ***')
      GO TO 6070

C      COMPUTE DEFAULT APRIOR VALUES FOR STANDARD CLASSIFIER
C      605 NORM = 1.0 / FLOAT(NOSUB2)
      DO 606 I=1,NOSUB2
        606 APRIOR(I) = NORM
      C      WRITE(6,6050)
      C      6050 FORMAT(15,'DEFAULT APRIORI PROBABILITY VALUES WILL BE USED. SURCL
        *ASS(I) = 1.0/(NO. OF SURCLASSES)')
      GO TO 6070
      6053 CONTINUE
      WRITE(6,6055)
      6055 FORMAT(15,'DEFAULT APRIORI PROBABILITY VALUES FOR SUBCLASS(I) = 1.
        *0/(NO OF CATEGORIES)*(NO. OF SUBCLASSES IN CATEGORY(J))')
      C      6070 APRFLG = NOSUR2
      WRITE(6,502)NFILE,NOFLD2,NOCLS2,NOSUB2,NOFET2
      607 IF (BMFLG .LE. 0) GO TO 700
      C      CALL WRBDM(BMATRX,BMCOMB,BMFEAT,FETVC2)
      C      700 CONTINUE
      C      IF (NOCHAN .NE. NOFET2) WRITE(6,800)
      800 FORMAT(15,'NO. OF CHANNELS REQUESTED FOR DATA TAPE AND NO. OF CHANNE
        *LS ON STAT// FIEL MUST BE EQUAL')
      IF (NOCHAN .NE. NOFET2) CALL CMERR
      -----
      WRITE FIRST RECORD ON THE CLASSIFICATION RESULTS OUTPUT FILE.
      MAPTAP
      -----
      HEADER RECORD NO. 1 FOR MAPTAP
```

SET03050
SET03060
SET03070
SET03080
SET03090
SET03100
SET03110
SET03120
SET03130
SET03140
SET03150
SET03160
SET03170
SET03180
SET03190
SET03200
SET03210
SET03220
SET03230
SET03240
SET03250
SET03260
SET03270
SET03280
SET03290
SET03300
SET03310
SET03320
SET03330
SET03340
SET03350
SET03360
SET03370
SET03380
SET03390
SET03400
SET03410
SET03420
SET03430
SET03440
SET03450
SET03460
SET03470
SET03480
SET03490
SET03500
SET03510
SET03520
SET03530
SET03540
SET03550
SET03560
SET03570
SET03580
SET03590
SET03600
SET03610
SET03620
SET03630
SET03640
SET03650
SET03660
SET03670
SET03680
SET03690
SET03700
SET03710
SET03720
SET03730
SET03740
SET03750
SET03760
SET03770
SET03780
SET03790
SET03800

FILE SETUP2

```

      WRITE(MAPTAP) (DATE(1),I=1,2),BMFLG,BMCOMB,BMFEAT,NOCLS2,
      *      NOFLD2,NOSUB2,NOFET2,TOTVT2,NOCAT,VARSZ2,(FETVC2(1),I=1,
      *      NOFET2)
      IF (NOCLS2.GT.1) GO TO 900
850    IF (ICK=LAHEAD) (FLDESC,VERTCS,FLDINF,NC)
      IF (ICK.NE.0) GO TO 850
      FLDPLG=1
      WRITE(6,5000)
5000    FORMAT(//,1X,'STATS INPUT FOR ONE CLASS. NO CLASSIFICATION'.
      1 ' ATTEMPTED.')
C
C
900    RETURN
      END

```

```

SET03810
SET03820
SET03830
SET03840
SET03850
SET03860
SET03870
SET03880
SET03890
SET03900
SET03910
SET03920
SET03930
SET03940

```


c c c

ORIGINAL PAGE IS
OF POOR QUALITY

THE UNIVERSITY OF CHICAGO

THW00010
 THH00020
 THH00030
 THH00040
 THH00050
 THH00060
 THH00070
 THH00080
 THH00090
 THH00100
 THH00110
 THH00120
 THH00130
 THH00140
 THH00150
 THH00160
 THH00170
 THH00180
 THH00190
 THH00200
 THH00210
 THH00220
 THH00230
 THH00240
 THH00250
 THH00260
 THH00270
 THH00280
 THH00290
 THH00300
 THH00310
 THH00320
 THH00330
 THH00340
 THH00350
 THH00360
 THH00370
 THH00380
 THH00390
 THH00400
 THH00410
 THH00420
 THH00430
 THH00440
 THH00450
 THH00460
 THH00470
 THH00480
 THH00490
 THH00500
 THH00510
 THH00520
 THH00530
 THH00540
 THH00550
 THH00560
 THH00570
 THH00580
 THH00590
 THH00600
 THH00610
 THH00620
 THH00630
 THH00640
 THH00650
 THH00660
 THH00670
 THH00680
 THH00690
 THH00700
 THH00710
 THH00720
 THH00730
 THH00740
 THH00750
 THH00760
 THH00770
 THH00780
 THH00790

$$\begin{array}{cccccc} 2,1 & & & & & \\ 3,1 & 3,2 & & & & \\ 4,1 & 4,2 & 4,3 & & & \\ 5,1 & 5,2 & 5,3 & 5,4 & & \\ 6,1 & 6,2 & 6,3 & 6,4 & 6,5 & \\ & \vdots & & & & \end{array}$$

COMPUTATION OF THE CLASS-PAIR THRESHOLDS

$$C = C2 - C1$$

(2) IF C2 .LE. C1 - (U1-U2)* X COV1**-1 X (U1-U2), THIJ= C2

FIND A NUMBER, $0 \leq x \leq 1$, SO THAT THE SOLUTION VECTOR, $H(x)$, OF THE SYSTEM OF EQUATIONS,

ALSO SATISFIES $G(H(X)) = C_2 - C_1$, WHERE

```

SUBROUTINE FALSJ DETERMINES 3-POINT INTERVALS IN THE RANGE.
0 .LE. X .LE. 1 , FITS A QUADRATIC Q(X) TO THE THREE POINTS AND
OBTAINS THE ROOT, X , FOR Q(X) = C2-C1. THE ROOT, X , OF THIS
QUADRATIC APPROXIMATION OF G( H(X) ) IS SENT TO FUNCTION G.

```

THRESH COMPUTES THE CLASS-PAIR THRESHOLD. THIS, BY OBTAINING FROM G THE EVALUATION OF $G(H(x))$ FOR THE x RETURNED BY FALSIFY :

INTEGER VARSZ2

12. DISPLAY PROCESSOR

ORIGINAL PAGE IS
OF POOR QUALITY

FILE DISPLAY

```

SUBROUTINE DSPLAY (ARRAY, TOP)
IMPLICIT INTEGER (A-H, O-Z)

DIMENSION ARRAY (1)

-----
CALL..    CALL DSPLAY (ARRAY, TOP)
ARGS..    ARRAY = SFE 'MONITOR'
          TOP   = SFE 'MONITOR'
PURPOSE.. COORDINATES ROUTINES FOR DISPLAYING CLASSIFICATION
          MAP AND PERFORMANCE TABLES.
-----
INCLUDE CMRK10, LIST
INCLUDE COMT10
COMMON/DISPL/ CATFLG, CATNAM (61), CLSNAM (61), SUBNAM (61), SUBNO (60),
*          SURCAT (60), CLSSUB (60), NOMAP, TOTVT3, NOSUB3,
*          PCFDKY, TSTKEY, TRNKEY, THRSKY, STATKY, EMPTRS, THRSVA,
*          PLTKEY, BMFLG, BMCOMB, BMFEAT, CDATE (2),
*          FLDSV2, FIELD2, VERTX2, FLDSV3, FIELD3, VERTX3, PCTID3,
*          THRS (60), SYMPTX (66), HIGH (60), CON (60),
*          FLCKEY, NOFLD2, NOFLD3, NOFET2, FETVC2 (30),
*          NOSUP2, NOTRFD, TOTVT2, NOCLS2,
*          KATNO (60), NOCAT, FILTER, MAPFMT,
*          DESKEY, DESUNI, DESOTH, CROP, ACROP, AOTHER, ATOTAL,
*          SITE (6), ANALYS (5), CAM (15), CRPKY, KEPPTS (60),
*          DOTKEY, DOTERR

COMMON BLOCK DISPL IS USED ONLY IN THE DISPLAY PROCESSOR

DEFINITIONS

CATFLG - FLAG INDICATING WHETHER OR NOT CATEGORY PERFORMANCE
        REPORTS MUST BE GENERATED.
CATNAM - NAMES OF CATEGORIES. READ FROM MAPTAP.
CLSNAM - NAMES OF CLASSES. READ FROM MAPTAP.
SUBNAM - NAMES OF SUBCLASSES. READ FROM MAPTAP.
SURCAT - SURCLASS-CATEGORY CORRESPONDENCE VECTOR
        (SURCAT(I)=M MEANS SUBCLASS I BELONGS TO CATEGORY M)
CLSSUB - SURCLASS-CLASS CORRESPONDENCE VECTOR.
        (CLSSUB(I)=M MEANS SUBCLASS I BELONGS TO CLASS M)
NOMAP  - TRIGGER INDICATING WHETHER OR NOT A MAP IS TO BE PRINTED
TOTVT3 - TOTAL NO. OF VERTICES IN INPUT TEST FIELDS.
NOSUB3 - NO. OF SUBCLASSES USED IN CLASSIFY PLUS ONE, FOR THE
        THRESHOLD CLASS.
PCFDKY - KEY INDICATING WHETHER OR NOT GROUND TRUTH PERFORMANCE
        REPORTS ARE TO BE PRINTED ON A PER FIELD BASIS.
TSTKEY - KEY INDICATING WHETHER OR NOT TEST FIELDS WERE INPUT.
TRNKEY - KEY INDICATING WHETHER OR NOT TRAINING FIELDS ARE TO
        BE OUTLINED.
CONTINUE
THRSKY - THRESHOLD KEY
        =1 APPLY CHI-SQUARE THRESHOLDS
        =2 APPLY EMPIRICAL THRESHOLDS
        =3 APPLY USER-INPUT THRESHOLDS
        =4 APPLY FISHER DISTRIBUTION THRESHOLD
        =0 NO THRESHOLDING
STATKY - KEY FOR PRINTING STATS FROM MAPTAP
EMPTRS - EMPIRICAL THRESHOLDING FLAG
THRSVA - USER-INPUT THRESHOLD VALUE FLAG
PLTKEY - FLAG FOR PRINTING CUMULATIVE HISTOGRAMS OF QUADRATIC
        FORM.
BMFLG  - FLAG INDICATING WHETHER OR NOT A B-MATRIX WAS
        APPLIED IN CLASSIFY.
BMCOMB - NO. OF LINEAR COMBINATIONS IN H-MATRIX
BMFEAT - NO. OF CHANNELS USED IN COMPUTING B-MATRIX

```

```

C* COATE - DATE OF CLASSIFICATION                                DSP00770
C* FLDSV2 - ADDRESS IN 'ARRAY' FOR TRAINING FIELD INFORMATION. DSP00780
C*          FOR EACH TRAINING FIELD 4 PIECES OF INFORMATION ARE DSP00790
C*          STORED - 1=FIELD NAME                                DSP00800
C*                  2=CLASS NO.                                DSP00810
C*                  3=SUBCLASS NO.                             DSP00820
C*                  4=NO. OF VERTICES                           DSP00830
C* FIELD2 - ADDRESS IN 'ARRAY' FOR RECTANGULAR AREA SURROUNDING DSP00840
C*          EACH TRAINING FIELD. FOR EACH TRAINING FIELD 5 PIECES DSP00850
C*          OF INFORMATION ARE STORED.                           DSP00860
C*                  1=LINE START                                DSP00870
C*                  2=LINE END                                  DSP00880
C* CONTINUE                                                     DSP00890
C*                  3=SAMPLE START                               DSP00900
C*                  4=SAMPLE END                                 DSP00910
C*                  5=POINTER INTO VERTEX ARRAY FOR VERTICES DSP00920
C*                    OF THIS FIELD.                             DSP00930
C* VERTX2 - ADDRESS IN 'ARRAY' FOR TRAINING FIELD VERTICES.    DSP00940
C* FLOSV3 - SAME AS FLOSV2 FOR TEST FIELDS                      DSP00950
C* FIELD3 - SAME AS FIELD2 FOR TEST FIELDS                      DSP00960
C* VERTX3 - SAME AS VERTX2 FOR TEST FIELDS                      DSP00970
C* PCTID3 - ADDRESS IN 'ARRAY' FOR PERFORMANCE TABLE.         DSP00980
C* THRES - THRESHOLD VALUES                                    DSP00990
C* SYMMTX - SYMBOLS FOR EACH SURCLASS, PLUS THRESHOLD SYMBOL    DSP01000
C*          AND OUTLINE SYMBOLS.                                DSP01010
C* HIGH - THRESHOLD REJECTION PERCENTAGE - EMPIRICAL OPTION    DSP01020
C* CON - CONSTANT FACTOR FROM PROBABILITY DENSITY FUNCTION     DSP01030
C*          FROM CLASSIFY. ONE FOR EACH SURCLASS.              DSP01040
C* FLOKEY - KEY INDICATING WHETHER GROUND TRUTH FIELDS ARE     DSP01050
C*          ASSOCIATED WITH CLASSES OR SUBCLASSES.             DSP01060
C* NOFLD2 - NO. OF TRAINING FIELDS                              DSP01070
C* NOFLD3 - NO. OF TEST FIELDS                                  DSP01080
C* NOFET2 - NO. OF CHANNELS USED IN CLASSIFICATION.            DSP01090
C* FETVC2 - CHANNELS USED IN CLASSIFICATION.                   DSP01100
C* NOSUR2 - NO. OF SUBCLASSES USED IN CLASSIFICATION.          DSP01110
C* NOTRFD - NO. OF GROUND TRUTH FIELDS FOR WHICH PERFORMANCE   DSP01120
C*          TABLES WILL BE MADE. EQUALS NOFLD3 OR NOFLD2.     DSP01130
C* TOTVT2 - TOTAL NO. OF VERTICES FOR TRAINING FIELDS.        DSP01140
C* NUCLS2 - NO. OF CLASSES USED IN CLASSIFICATION.            DSP01150
C* KATNO - CLASS - CATEGORY CORRESPONDENCE VECTOR             DSP01160
C* CONTINUE                                                     DSP01170
C*          (KATNO(I)=M MEANS CLASS I IS IN CATEGORY M)        DSP01180
C* NOCAT - NO. OF CATEGORIES.                                    DSP01190
C* FILTER - FLAG FOR SPATIAL FILTERING OPTION.                 DSP01200
C* MAPFMT - FORMAT FOR OUTPUT MAP TAPE                          DSP01210
C* DESKEY - KEY INDICATING WHETHER OR NOT DESIGNATED FIELDS WERE IN DSP01220
C* DESUNI - NO. FOR DESIGNATED UNIDENTIFIABLE (NOSUB2+5)      DSP01230
C* DESOTH - NO. FOR DESIGNATED OTHER (NOSUR2+6)               DSP01240
C* CROP - NAME OF CROP FOR WHICH INTENSIVE TEST SITE SUMMARY   DSP01250
C*          REPORT IS TO BE PRINTED. CROP IS TO BE COMPARED WITH OT DSP01260
C* ACROP - ACRES OF 'CROP' - USER INPUT                        DSP01270
C* AOTHER - ACRES OF 'OTHER' - USER INPUT                      DSP01280
C* ATOTAL - TOTAL ACRES IN CLASSIFIED SEGMENT                  DSP01290
C* SITE - NAME OF SITE (CLASSIFIED SEGMENT)                    DSP01300
C* ANALYS - NAME OF ANALYST PERFORMING STUDY                   DSP01310
C* CAMS - NAME OF PROCEDURE CONFIGURATION USED IN STUDY        DSP01320
C* CRPKEY - KEY FOR GENERATING INTENSIVE TEST SITE SUMMARY REPORT DSP01330
C* KPPPTS - TOTAL NUMBER PIXELS IN EACH SUBCLASS              DSP01340
C*                                                              DSP01350
C* DOTKEY - KEY INDICATING WHETHER OR NOT DOT DATA CLASSIFICATION DSP01360
C* PERFORMANCE SUMMARIES ARE TO BE PROCESSED: DOTKEY = 0, NO DOT DSP01370
C* DATA PROCESSING; DOTKEY .GT. 0, DOT PERFORMANCE SUMMARIES DSP01380
C* ARE PROVIDED (CHANGED TO INDICATE LIST PROCESSING           DSP01390
C* INSTEAD OF DOT PROCESSING ON MAY 1979)                      DSP01400
C*                                                              DSP01410
C* DOTERR - USE OF THIS FLAG REMOVED MAY 1979                 DSP01420
C*          FLAG NOT NEEDED WHEN LIST SUBSTITUTED FOR DOT PROCESSING DSP01430
C* CONTINUE                                                     DSP01440
C*                                                              DSP01450
C* C$END                                                         DSP01460
C*                                                              DSP01470
C*                                                              DSP01480
C* SETUP3 WILL READ FIRST 2 RECORDS FROM MAPTAP, AND CALL REDIF3 DSP01490
C* TO READ IN CONTROL CARDS. ALL OF THE PARAMETERS IN COMMON BLOCK DSP01500
C* DISPL ARE INITIALIZED BEFORE RETURNING TO THIS ROUTINE IN ADDITION DSP01510
C* TO TRAINING AND/OR TEST FIELD DEFINITIONS WILL BE STORED IN 'ARRAY' DSP01520

```

FILE DSPLAY

```

C*
C*
C***      THIS ADDED OR CHANGED NOV.13,1978 TO INCLUDE LIST PROS.
C
      DIMENSION DESSAV(4,50),DESFLD(5,50),DESVR(1100)
      REAL ALP(2)
      STOP=0
      CALL SETUP3(ARRAY, TOP, GTUNIT, GTFILE, AIUNIT, AIFILE,
*      PPUNIT, PPFILE, NAMECT, ALP, DESSAV, DESFLD, DESVR, NOFLD4, STOP)
      IF (STOP.EQ.0) GO TO 5
      WRITE(6,6100)
6100  FORMAT(//,1X,'NO MAP AVAILABLE FROM CLASSIFICATION PROCESSOR.')
```

DSP01530
DSP01540
DSP01550
DSP01560
DSP01570
DSP01580
DSP01590
DSP01600
DSP01610
DSP01620
DSP01630
DSP01640
DSP01650
DSP01660
DSP01670
DSP01680
DSP01690
DSP01700
DSP01710
DSP01720
DSP01730
DSP01740
DSP01750
DSP01760
DSP01770
DSP01780
DSP01790
DSP01800
DSP01810
DSP01820
DSP01830
DSP01840
DSP01850
DSP01860
DSP01870
DSP01880
DSP01890
DSP01900
DSP01910
DSP01920
DSP01930
DSP01940
DSP01950
DSP01960
DSP01970
DSP01980
DSP01990
DSP02000
DSP02010
DSP02020
DSP02030
DSP02040
DSP02050
DSP02060
DSP02070
DSP02080
DSP02090
DSP02100
DSP02110
DSP02120
DSP02130
DSP02140
DSP02150
DSP02160
DSP02170
DSP02180
DSP02190
DSP02200
DSP02210
DSP02220
DSP02230
DSP02240

```

      GO TO 999
C*
C*      DSPLY1 WILL READ NEXT 2 RECORDS FROM MAPTAP AND PRINT THE
C*      STATISTICS IF REQUESTED.
C*
C*      CALL DSPLY1
C
C***      CODE ADDED NOV. 13,1978 TO INCLUDE LIST PROCESSING
C****     CODE CHANGED MAY 1979 TO SUBSTITUTE LIST FOR DOTS
C
      IF (DOTKEY.NE.0) GO TO 30
      IF (EMPTRS.NE.2.AND. PLTKEY.NE.1) GO TO 30
C*
C*      EMTHRS COMPUTES AND PLOTS THE HISTOGRAM OF THE QUADRATIC FORM
C*      FOR THE CORRECTLY CLASSIFIED PIXELS WITHIN THE TRAINING OR TEST
C*      FIELDS.
C*
      IF (FLDKEY.EQ.1) GO TO 10
      WRITE(6,100)
      GO TO 30
10  CONTINUE
      IF (TSTKEY.EQ.1) CALL EMTHRS (ARRAY (FLDSV3), ARRAY (FIELD3),
*      ARRAY (VERTX3), NOFLD3)
      IF (TSTKEY.NE.1) CALL EMTHRS (ARRAY (FLDSV2), ARRAY (FIELD2),
*      ARRAY (VERTX2), NOFLD2)
30  CONTINUE
C-
C-      TEST THRSKY = 4 FOR FISHER F-DISTRIBUTION THRESHOLDS
C-      CALL FDIST TO COMPUTE AND STORE THRESHOLDS
C-
      IF (THRSKY.EQ.4) CALL FDIST
C-
C*
C*      DSPLY2 PRINTS THE MAP AND CALLS PCT TO BUILD PERFORMANCE TABLES.
C*
      CALL DSPLY2 (ARRAY (FLDSV2), ARRAY (FIELD2), ARRAY (VERTX2),
*      ARRAY (FLDSV3), ARRAY (FIELD3), ARRAY (VERTX3),
*      ARRAY (PCTID3), GTUNIT, GTFILE,
*      AIUNIT, AIFILE, PPUNIT, PPFILE, NAMECT, ALP,
*      DESSAV, DESFLD, DESVR, NOFLD4)
C*
C*      IF DOT DATA PROCESSING WAS REQUESTED, THE PERFORMANCE
C*      TABLES WERE PERFORMED IN DSPLY2
C
C***      CODE ADDED NOV 13,1978 TO INCLUDE LIST PROCESSING
C***      CODE CHANGED MAY 1979 TO SUBSTITUTE LIST FOR DOT PROCESSING
C
      IF (DOTKEY.GT.0) GO TO 99
C*
C*      PRTPCT PRINTS THE PERFORMANCE TABLES
C*
      IF (TSTKEY.NE.1) CALL PRTPCT (ARRAY (FLDSV2), ARRAY (PCTID3), NOFLD2)
99  IF (TSTKEY.EQ.1) CALL PRTPCT (ARRAY (FLDSV3), ARRAY (PCTID3), NOFLD3)
      WRITE(6,6000)
6000  FORMAT(1H1X,' ***** DISPLAY COMPLETED *****')
100  FORMAT(//, ' **DISPLAY** - FIELDS MUST BE DEFINED FOR SUBCLASSES FOR
*      EMPIRICAL THRESHOLDS')
999  RETURN
      END
```

FILE: CHI

| | | |
|--------|---|----------|
| C----- | REAL FUNCTION CHI(X,N,IFLAG) | CHI00010 |
| C----- | TO COMPUTE THE VALUE OF THE CHI-SQUARED DISTRIBUTION WITH N-D.F | CHI00020 |
| C----- | IF(X.GT.0.0) GO TO 5 | CHI00030 |
| | CHI=0.0 | CHI00040 |
| | RETURN | CHI00050 |
| C----- | CHCK TO SEE IF THE DEGREES OF FREEDOM IS EVEN | CHI00060 |
| C----- | 5 IF(MOD(N,2).EQ.0) GO TO 1 | CHI00070 |
| C----- | CALCULATION OF CHI FOR 1 DEGREE OF FREEDOM | CHI00080 |
| C----- | G=SQRT(X) | CHI00090 |
| | CHI=2.0*RNORM(G)-1.0 | CHI00100 |
| | G=G/1.25331414 | CHI00110 |
| | IN=3 | CHI00120 |
| | GO TO 2 | CHI00130 |
| C----- | CALCULATION OF CHI FOR 2 DEGREES OF FREEDOM | CHI00140 |
| C----- | 1 IN=4 | CHI00150 |
| | G=X/2.0 | CHI00160 |
| | IF(ABS(G).GT.88.027) GOTO 4 | CHI00170 |
| | CHI=1.0-EXP(-G) | CHI00180 |
| | 2 IF(N.LT.3) RETURN | CHI00190 |
| | IF(ABS(X/2.0).GT.88.027) GOTO 4 | CHI00200 |
| | G=G*EXP(-X/2.0) | CHI00210 |
| C----- | CALCULATION OF CHI FOR N-GT-2 DEGREES OF FREEDOM | CHI00220 |
| C----- | DO 3 I=IN,N*2 | CHI00230 |
| | CHI=CHI-G | CHI00240 |
| | G=G*X/I | CHI00250 |
| | CALL OVERFL(INDCT) | CHI00260 |
| | IF(INDCT.EQ.1) GOTO 4 | CHI00270 |
| | 3 CONTINUE | CHI00280 |
| | RETURN | CHI00290 |
| | 4 IFLAG=1 | CHI00300 |
| | RETURN | CHI00310 |
| | END | CHI00320 |
| | | CHI00330 |
| | | CHI00340 |
| | | CHI00350 |
| | | CHI00360 |
| | | CHI00370 |
| | | CHI00380 |
| | | CHI00390 |
| | | CHI00400 |
| | | CHI00410 |
| | | CHI00420 |

FILE: CHIN

ORIGINAL PAGE IS
OF POOR QUALITY

```

FUNCTION CHIN(ALPHA,N,IFLAG)
DIMENSION H(7), D(15)
EQUIVALENCE (H(7), D(15))
DATA D/- .7080,-.4020,-.1980,-.0600,.0060,.0360,.0360,
1 .0120,-.0180,-.0360,-.0300,.0120,.1020,.2580,.4920/
IFLAG=0
IF(N.EQ.1) CHIN=(TINORM((1.-ALPHA/2.),IFLAG))
IF(IFLAG.EQ.1) GOTO 10
CHIN=CHIN**2
IF(N.EQ.2) CHIN = -2. * ALOG(ALPHA)
IF(N.LE.2) RETURN
X=TINORM(1.-ALPHA,IFLAG)
IF(IFLAG.EQ.1) GOTO 10
I=2.*X
IF(I.LT.-7) I=-7
IF(I.GT.6) I=6
Y=(H(I)+(2.*X-I)*(H(I+1)-H(I)))/N
CHIN=2./(9.*N)
X=N*(1.-CHIN+(X-Y)*SQRT(CHIN))**3
IF((N.LE.55.AND.X.LT.0.) ) X=N*(1.-CHIN+X*SQRT(CHIN))**3
IF(N.LE.55) GO TO 1
IF(X.GT.176.16) GOTO 10
CHIN=X
4 RETURN
1 IC=0
IE=N-2
IR=3
G=2.5066283
IF(MOD(N,2).EQ.1) GO TO 2
IR=2
G=2.
2 IF(IE.LE.1) GO TO 11
DO 3 I=1,IE,2
3 G=G*I
11 N2 = (N-2)/2
N3 = N-2-N2
SQX = SQRT(X)
CHA = ((1.-CHI(X,N,IFLAG)-ALPHA)*G)/(SQX**N2)
IF(IFLAG.EQ.1) GOTO 10
CHR = EXP(X/2.)/(SQX**N3)
CHIN = X + CHA*CHR
IF(ABS(X-CHIN)/AMAX1(X,CHIN).LT.5.E-06) GO TO 4
IF(IC.GT.200) RETURN
IC=IC+1
X=CHIN
IF(X.GT.176.16) GOTO 10
GO TO 11
10 IFLAG=1
RETURN
END

```

CHI00010
CHI00020
CHI00030
CHI00040
CHI00050
CHI00060
CHI00070
CHI00080
CHI00090
CHI00100
CHI00110
CHI00120
CHI00130
CHI00140
CHI00150
CHI00160
CHI00170
CHI00180
CHI00190
CHI00200
CHI00210
CHI00220
CHI00230
CHI00240
CHI00250
CHI00260
CHI00270
CHI00280
CHI00290
CHI00300
CHI00310
CHI00320
CHI00330
CHI00340
CHI00350
CHI00360
CHI00370
CHI00380
CHI00390
CHI00400
CHI00410
CHI00420
CHI00430
CHI00440
CHI00450
CHI00460
CHI00470
CHI00480
CHI00490
CHI00500

FILE: DESIG

```
      SUBROUTINE DESIG(LINE,IR,FLDSAV,FIELD,VERTEX,NOFLD,  
      SAMSTR,SAMEND,SAMINC)  
      IMPLICIT INTEGER(A-Z)  
      DIMENSION IR(1),FIELD(5,NOFLD),FLDSAV(4,NOFLD),VERTEX(1),FL(22)  
C*  
C* THIS ROUTINE SETS THE IR ARRAY FOR DESIGNATED FIELDS  
C*  
      DO 50 I=1,NOFLD  
      IF(LINE.LT.FIELD(1,I))GO TO 50  
      IF(LINE.GT.FIELD(2,I))GO TO 50  
      IF(FIELD(3,I).GT.SAMEND)GO TO 50  
      IF(FIELD(4,I).LT.SAMSTR)GO TO 50  
C*  
C* FOUND A DESIGNATED FIELD ON THIS LINE  
C*  
      NV=FLDSAV(4,I)  
      IPT=FIELD(5,I)  
      ID = FLDSAV(2,I)  
      CALL FDLINT(VERTEX(IPT),NV,FL,LINE,SAMPS,NI)  
      DO 20 J=1,NI,2  
      IR = (FL(J)-SAMSTR)/SAMINC+1  
      IE = (FL(J+1)-SAMSTR)/SAMINC+1  
      IF(MOD(SAMSTR,SAMINC).NE.MOD(FL(J),SAMINC)) IB=IB+1  
      IF(IR.GT.IE)GO TO 20  
      DO 10 K=IB,IE  
10  IR(K)=ID  
20  CONTINUE  
50  CONTINUE  
      RETURN  
      END
```

ORIGINAL PAGE IS
OF POOR QUALITY

FILE: DISTCV

```

SUBROUTINE DISTCV(DSFUNC,TOTPTS,RANGE)
IMPLICIT INTEGER(A-Z)
*****C
DISTCV PLOTS THE DISTRIBUTION AND CHI SQUARE CURVES AND COMPUTES
THE EMPIRICAL THRESHOLD VALUES
*****C
REAL Q,DSFUNC,DISTVL,THRESH,REJECT,PCTREJ,THRES,CHISQ1
REAL REJFCT,CHISQ,INC,CHIN
DIMENSION DSFUNC(RANGE,60),TOTPTS(1),THRES(60),MINM(60),
SYNPLS(100),FIELDS(2),FIELD1(2),PCTREJ(60)
DIMENSION CHISQ(100)
C
INCLUDE CMRK10.LIST
COMMON/DISPL/CATFLG,CATNAM(61),CLSNAM(61),SUBNAM(61),SUBNO(60),
SURCAT(60),CLSSUB(60),NOMAP,TOTVT3,NOSUB3,
PCFOKY,ISTKEY,TRNKEY,THRSKY,STATKY,EMPTR5,THRSVA,
PLTRKY,RMFLG,RMCUMH,RMFFAT,CUATE(2),
FLOSV2,FIELD2,VERTX2,FLOSV3,FIELD3,VERTX3,PCTID3,
THRES(60),SYNMTX(66),HIGH(60),CON(60),
FLOKEY,NOFLD2,NOFLD3,NOFET2,FETVC2(30),
NOSUB2,NOTHFD,TOTVT2,NOCLS2,
KATNO(60),NOCAT,FILTER,MAPFMT,
DESKEY,DESUNI,DESOTH,CROP,ACROP,AOTHER,ATOTAL,
SITE(6),ANALYS(5),CAM(15),CRPKY,KEPPTS(60),
DOTKEY,DOTERR
CSEND
DATA FIELD/'TEST',FIELDS/'TRAI','NING'/
DATA BLANK/'',MXQUAD/20/
DATA ASTK/'%',DOLLAR/'$'/
EQUIVALENCE (HIGH(1),PCTREJ(1))
DO 5 I=1,2
5 FIFLD1(I)=BLANK
DO 10 L=1,99
REJFCT=1.-FLOAT(L)/100.0
CHISQ1=CHIN(REJFCT,NOFET2,FLAG)
IF(FLAG.EQ.1) GOTO 13
CHISQ(L)=CHISQ1+.05
GO TO 10
13 WRITE(6,140) REJFCT,CHISQ1
140 FORMAT(' OVERFLOW',2X,F5.2,3X,F15.5)
10 CONTINUE
CALL SETMRG(66,0.66)
DO 200 J=1,NOSUB2
TOTPTS(J)=0
DO 20 M=1,RANGE
TOTPTS(J)=DSFUNC(M,J)+TOTPTS(J)
20 CONTINUE
IF (TOTPTS(J).EQ.0) GO TO 200
DO 40 M=1,RANGE
MM=M-1
IF (MM.EQ.0) DSFUNC(MM+1,J)=(DSFUNC(MM+1,J)/TOTPTS(J))*100
IF (MM.EQ.0) GO TO 40
DSFUNC(M,J)=DSFUNC(MM,J)+(DSFUNC(M,J)/TOTPTS(J))*100
40 CONTINUE
C
C
FIND MINIMUM M FOR WHICH DSFUNC(M,J).GT.1-100*PCTREJ
DISTVL=(1.-PCTREJ(J))*100
MXREJT=DISTVL+.5
DO 50 M=1,RANGE
IF (DISTVL.LE.DSFUNC(M,J)) MINM(J)=M
IF (DISTVL.LE.DSFUNC(M,J)) GO TO 55
50 CONTINUE
55 CONTINUE
C
C
THRESHOLD
REJECT=PCTREJ(J)*100
IF (ISTKEY.EQ.1) FIELD1(1)=FIELD
IF (ISTKEY.EQ.0) FIELD1(1)=FIELDS(1)
IF (TRNKEY.EQ.0) FIELD1(2)=FIELDS(2)
THRESH(J)=(0.1*MINM(J))*2
C
C
PRINT HEADING
CLASNO=J

```

FILE: DISTCV

```

WRITE(6,100)SUBNAM(J)
100 FORMAT(1H1,///T52,'DISTRIBUTION CURVE FOR SURCLASS ',A4)
WRITE(6,105)THRES(CLASNO),REJECT,(FIELD1(I),I=1,2),THRESH(J)
105 FORMAT(///T10,'CHI SQUARE THRESHOLD = ',F5.2,T53,'EMPIRICAL T
*HRESHOLD FROM',T91,'USER REJECTION PERCENTAGE = ',F4.1/T53,
* 2A4,' FIELDS = ',F5.2)
WRITE(6,110)
110 FORMAT(///T15,10(1H0,9X),1H1,/,T15,'0',9X,'1',9X,'2',9X,'3',9X,
*  '4',9X,'5',9X,'6',9X,'7',9X,'8',9X,'9',9X,'0',/
*  T15,11(1H0,9X)/
*  T11,'0.0',1X,101(1H+))
INC = FLOAT(MXQUAD)/FLOAT(RANGE)
II = 1
N = 0
DO 90 L=1,RANGE
Q = FLOAT(MXQUAD*L) / RANGE
N = N + 1
DO 60 M=1,100
60 SYMBS(M) = BLANK
CHI SQUARE CURVE
65 IF ( II .GT. MXREJT ) GO TO 75
IF ( CHISQ(II) .GE. (Q+INC) ) GO TO 75
SYMBS(II) = ASTK
II = II + 1
GO TO 65
75 CONTINUE
DISTRIBUTION CURVE
PERCNT = DSFUNC(L,J)
IF (PERCNT .EQ. 0) GO TO 77
IF (SYMBS(PERCNT) .NE. BLANK) SYMBS(PERCNT) = DOLLAR
IF (SYMBS(PERCNT) .NE. BLANK) GO TO 77
76 SYMB=J
SYMBS(PERCNT) = SYMMTX(SYMB)
77 CONTINUE
IF (N .EQ. 5) GO TO 80
WRITE(6,120) (SYMBS(K),K=1,100)
120 FORMAT(T15,'+',100A1)
GO TO 90
80 WRITE(6,130) Q, (SYMBS(K),K=1,100)
130 FORMAT(T10,F4.1,1X,'+',100A1)
N = 0
90 CONTINUE
WRITE(6,150) SYMMTX(SYMB)
150 FORMAT(///T10,'NOTE : ',A1,' - CLASS DISTRIBUTION CURVE',/T18,
*  ' - CHI SQUARE DISTRIBUTION CURVE',/T18,'S - INTERSECTION OF CURV
*  FS')
IF (EMPTRS .EQ. 0) GO TO 200
THRES(J)=THRESH(J)
200 CONTINUE
CALL SETMRG(66,4,62)
RETURN
END

```

DIS00800
DIS00810
DIS00820
DIS00830
DIS00840
DIS00850
DIS00860
DIS00870
DIS00880
DIS00890
DIS00900
DIS00910
DIS00920
DIS00930
DIS00940
DIS00950
DIS00960
DIS00970
DIS00980
DIS00990
DIS01000
DIS01010
DIS01020
DIS01030
DIS01040
DIS01050
DIS01060
DIS01070
DIS01080
DIS01090
DIS01100
DIS01110
DIS01120
DIS01130
DIS01140
DIS01150
DIS01160
DIS01170
DIS01180
DIS01190
DIS01200
DIS01210
DIS01220
DIS01230
DIS01240
DIS01250
DIS01260
DIS01270
DIS01280
DIS01290
DIS01300
DIS01310
DIS01320
DIS01330
DIS01340
DIS01350

ORIGINAL PAGE IS
OF POOR QUALITY

FILE: DSPLY1

```
C      SUBROUTINE DSPLY1
C      IMPLICIT INTEGER (A-M,0-Z)
C      -----
C      PURPOSE.. READS THE STATISTICS FROM 'MAPTAP'
C      -----
C      INCLUDE COMRK6.LIST
C      REAL CON,DET(60)
C      INCLUDE CMH10.LIST
C      COMMON/GLOBAL/HEAD(63),MAPTAP,DATAP,SAVTAP,BMFILE,BMKEY,
C      HISFIL,HISKEY,TRFORM,ER1PTP,ERPKEY,MAPUNT,NOFILE,
C      DRUMAD,DRMWDN,PAGSIZ,DATFIL,STAFIL,ASAV,ASAVFL
C      ,NHSTUN,NHSTFI,SCTRUN,MAPFIL
C      ,DOTUNT,DOTFIL,NCHPAS,TRNSFL,BMTRFL,HISTFL,PCHUNT,
C      CPDUNT,PRUNT,RANDIU
C      COMMON/DISPL/CATFLG,CATNAM(61),CLSNAM(61),SUBNAM(61),SUBNO(60),
C      SURCAT(60),CLSSUB(60),NOMAP,TOTVT3,NOSUB3,
C      PCFDKY,TSTKEY,TRNKEY,THRSKY,STATKY,EMPTRS,THRSVA,
C      PLTKEY,BMFLG,BMCOMB,BMFEAT,CDATE(2),
C      FLDSV2,FLDID2,VERTX2,FLDSV3,FLDID3,VERTX3,PCTID3,
C      THRES(60),SYMMTX(66),HIGH(60),CON(60),
C      ,FLOKEY,NOFLD2,NOFLD3,NOFET2,FETVC2(30)
C      ,NOSUH2,NOTREFD,TOTVT2,NOCLS2
C      ,KATNO(60),NOCAT,FILTER,MAPFMT
C      ,DESKEY,DESUNI,DESOTH,CROP ,ACROP,AOTHER,ATOTAL
C      ,SITE(6),ANALYS(5),CAM(15),CRPKEY,KEPPTS(60)
C      ,DOTKEY,DOTERR
CSEND  DIMENSION STORAGE(9500)
DATA SIZE/9500/
DATA ACOFOR/'4'//,BCDWO/'2'//,DASH/'----'//
C      -----
C      RETRIEVE AND PRINT THE COVARIANCE AND MEAN
C      -----
C 200 IF( BMFLG .LE. 0) GO TO 202
C      NOFET2 = BMCOMB
C 202 CV=1
VARSZ2=NOFET2*(NOFET2+1)/2
MN=CV + VARSZ2*NOSUH2
IF(MN + NOSUH2*NOFET2 .GT. SIZE)GO TO 180
GO TO 190
C 170 CONTINUE
CALL DSPLIA(STORAGE(CV),STORAGE(MN),VARSZ2,NOFET2,NOSUB2)
RETURN
C 190 WRITE(6,1A1)
181 FORMAT(' NOT ENOUGH STORAGE FOR COVARIANCE MATRICES - DSPLY1')
CALL CMERR
CONTINUE
RETURN
C      END --- SUBROUTINE DSPLY1
C      -----
C      INTERNAL SUBROUTINE USPLIA
C      SUBROUTINE USPLIA(COVMTX,AVEMTTX,VARSZ2,NOFET2,NOCLS2)
COVMTX(1,1)=STORAGE(1)
AVEMTX(1,1)=STORAGE(MN-1+1+(J-1)*NOFET2
C 190 CONTINUE
C      READ ORIGINAL COVARIANCE AND MEANS MATRIX FOR EACH CLASS
C      (H-TRANSFORMED IF H-MATRIX WAS APPLIED IN SCLASSIFY -
C      BMFLG .GT. 0 IF SO )
```

FILE: DPLY1

```

DUMMY=NOCLS2
NOCLS2=NOCLM2
NS = VARSZ2 * NOCLS2
NSS = NOFET2 * NOCLS2
C READ(MAPTAP) (STORAG(CV+I-1),I=1,NS), (STORAG(MN+I-1),I=1,NSS)
C IF (STATKY.EQ.0) GO TO 290
C CNT = 7*(5+3*2*NOFET2)*((NOFET2+11)/12)
C CNT = PAGESZ/CNT
C INC = CNT
C DO 280 ICLAS=1,NOCLS2
C IF (INC.LT.CNT) GO TO 210
C WRITE (6,HEAD)
C INC = 0
210 WRITE (6,220) SUBNAM(ICLAS),SYMMTX(ICLAS),(DASH,I=1,5)
220 FORMAT(//,'SUBCLASS',A4,' REPRESENTED BY SYMBOL - ',A1 /
* 1X,3A4,A1,T44,A3/)
C DO 230 LOC=1,NOFET2,12
C STOP = LOC+11
C IF (STOP.GT.NOFET2) STOP = NOFET2
C NS=MN-1+(ICLAS-1)*NOFET2
C WRITE (6,240) (STORAG(NS+I),I=LOC,STOP)
230 CONTINUE
260 FORMAT('0 MEAN:',3X,12F9.2)
C WRITE (6,2401) DASH
2601 FORMAT(1X,A4/)
C IF (RMFLG.GT.0) GO TO 271
C WRITE (6,270) (DASH,I=1,5)
C GO TO 272
270 FORMAT('COVARIANCE MATRIX:' / 1X,5A4)
271 WRITE (6,2711) (DASH,I=1,9)
2711 FORMAT('COVARIANCE MATRIX (B-TRANSFORMED) ' / 1X,8A4,A3)
272 NS=1+(ICLAS-1)*VARSZ2
C CALL WRMTX(STORAG(NS),NOFET2,BCDWO)
C INC = INC+1
280 CONTINUE
C READ COVARIANCE MATRIX ( AFTER CHOLESKY FACTORIZATION),
C PROBABILITY DENSITY FUNCTION CONSTANTS, CON , AND COVARIANCE
C MATRIX DETERMINANT, DET , FOR EACH CLASS
C -----
290 CONTINUE
C NS = VARSZ2 * NOCLS2
C READ(MAPTAP) (STORAG(CV+I-1),I=1,NS), (CON(I),I=1,NOCLS2), (DET(I),
* I=1,NOCLS2)
C IF (STATKY.EQ.0) GO TO 330
C CNT = 11*(3+2*NOFET2)*((NOFET2+11)/12)
C CNT = PAGESZ/CNT
C INC = CNT
C DO 310 I=1,NOCLS2
C IF (INC.LT.CNT) GO TO 300
C WRITE (6,HEAD)
C INC = 0
300 WRITE (6,320) SUBNAM(I),SYMMTX(I),DET(I),CON(I)
C NS = 1+(I-1)*VARSZ2
C CALL WRMTX(STORAG(NS),NOFET2,BCDFOR)
C INC = INC+1
310 CONTINUE
320 FORMAT(//1X,'MULTISPECTRAL CHARACTERISTICS FOR SUBCLASS ',A4,'
1 REPRESENTED BY SYMROL ',A1/1H0,' DETERMINANT = ',F18.8/1H0,' CONSTANDSP0
2 TERM = ', F10.4//1H0,' COVARIANCE MATRIX (CHOLESKY DECOMPOSITION)
1 :')
C GO HOME
C -----
330 CONTINUE
C NOCLS2=DUMMY
C GO TO 170
C END

```

DSP00800
 DSP00810
 DSP00820
 DSP00830
 DSP00840
 DSP00850
 DSP00860
 DSP00870
 DSP00880
 DSP00890
 DSP00900
 DSP00910
 DSP00920
 DSP00930
 DSP00940
 DSP00950
 DSP00960
 DSP00970
 DSP00980
 DSP00990
 DSP01000
 DSP01010
 DSP01020
 DSP01030
 DSP01040
 DSP01050
 DSP01060
 DSP01070
 DSP01080
 DSP01090
 DSP01100
 DSP01110
 DSP01120
 DSP01130
 DSP01140
 DSP01150
 DSP01160
 DSP01170
 DSP01180
 DSP01190
 DSP01200
 DSP01210
 DSP01220
 DSP01230
 DSP01240
 DSP01250
 DSP01260
 DSP01270
 DSP01280
 DSP01290
 DSP01300
 DSP01310
 DSP01320
 DSP01330
 DSP01340
 DSP01350
 DSP01360
 DSP01370
 DSP01380
 DSP01390
 DSP01400
 DSP01410
 DSP01420
 DSP01430
 DSP01440
 DSP01450
 DSP01460
 DSP01470
 DSP01480
 DSP01490
 DSP01500
 DSP01510
 DSP01520
 DSP01530
 DSP01540
 DSP01550

FILE DSPLY2

~~12-11~~

190

FILE DSPLY2:

```

DATA FORMAT/'UNIV','ERSA',.1L  'LARS','YS I',.1I  '
REAL TOTALS(66),VR,THRES
DIMENSION TTOL(66)
-----
TRNNO=NOSUR3+1
TSTNO=NOSUR3+2
SET POINTERS FOR SYMBOLS ARRAY AND TOTALS ARRAY
DUPNO = NOSUR3+3
DESUNI= NOSUR3+4
DESOTH= NOSUR3+5
FLAG USED IN DOTPCT TO INITIALIZE PCTAB=0
PCTKEY=0
C*** CODE ADDED NOV 13,1978 TO INCLUDE LIST PROCESSING
      FLD CNT = 0
      IF (NOTKEY.EQ.0) GO TO 17
      DO 16 I = 1,209
      LR = (I - 1)/19
      LR = (LR + 1)*10
      LS = (LR - 1)/10
      LS = 10*(I - (LS*19))
      TRNVER(1,I) = LS
      TRNVER(2,I) = LR
16 CONTINUE
CONTINUE
DO 10 I=1,NOSUR3
DO 10 J=1,NOTRFO
10 PCTAB(J,I) = 0
-----
PRINT OUT HEADING
-----
20 READ(MAPTAP)FLDINF,PTS,LINES,FLDESC
IF (PTS.GT. 1000) WRITE(6,22)
IF (PTS.GT. 1000) STOP
22 FORMAT(' DISPLAY WILL ACCEPT ONLY 1000 PTS/SCAN LINE')
ISTR=SAMSTR
IEND=SAMEND
IF (PTS.EQ.0) GO TO 310
MDREC = 1
DO 25 I=1,DESOTH
TTOL(I)=0
25 TOTALS(I)=0.0
-----
PRINT OUT THE COLUMN NUMBERS
-----
30 J = 0
CALL SETMRG(68,0,68)
IF (NOMAP.EQ.0)GO TO 85
WTFG=1
GO TO 370
31 WTFG=1
GO TO 510
85 CONTINUE
SPKNT=0
COUNT=0
DRUMLN=0
LAST=0
START=.FALSE.
FULL=.FALSE.
I1=1
I2=2
I3=3
J=1
AURES=DRUMAD

```

ORIGINAL PAGE IS
OF POOR QUALITY

FILE DSPLY2

```

91 READ(MAPTAP) ILINE(J),(IR(I,J),I=1,PTS),(VR(I),I=1,PTS)      DSP01530
IF(ILINE(J).EQ.0)GO TO 105                                         DSP01540
C*                                                                    DSP01550
C* IF DESIGNATED FIELDS HAVE BEEN INPUT, SET THE IR ARRAY FOR PIXELS DSP01560
C* IN THOSE FIELDS.                                                DSP01570
IF(DESKEY.EQ.1)CALL DESIG(ILINE(J),IR(I,J),DESSAV,DESFLD,DESVR, DSP01580
* NOFLD4,SAMSTR,SAMEND,SAMINC)                                     DSP01590
DO 100 I=1,PTS                                                     DSP01600
L = IR(I,J)                                                         DSP01610
IF(L.EQ.0)GO TO 100                                                 DSP01620
IF(L.EQ.NOSUB3)GO TO 92                                             DSP01630
IF(L.GT.NOSUB3)GO TO 95                                             DSP01640
C*                                                                    DSP01650
C* THRESHOLDING                                                    DSP01660
C* THE VALUE OF THE QUADRATIC FORM MUST BE EXTRACTED FROM VR FOR THRE DSP01670
C* Q = -2*VR - CON                                                 DSP01680
Q = -2*VR - CON                                                    DSP01690
C*                                                                    DSP01700
IF(THRSKY.EQ.0)GO TO 95                                             DSP01710
IF((-2.*VR(I)-CON(L)).LT. THRES(L))GO TO 95                        DSP01720
92 CONTINUE                                                         DSP01730
C*                                                                    DSP01740
C* TTOL = TOTAL PIXELS THRESHOLDED, BY SUBCLASS                   DSP01750
C*                                                                    DSP01760
TTOL(L) = TTOL(L) + 1                                              DSP01770
IR(I,J) = NOSUB3                                                  DSP01780
L = NOSUB3                                                         DSP01790
C*                                                                    DSP01800
C* TOTALS = TOTAL NO. PIXELS CLASSIFIED INTO EACH SUBCLASS,     DSP01810
C* INCLUDING THRESHOLDED AND DESIGNATED OTHER, DESIGNATED UNIDENT DSP01820
C*                                                                    DSP01830
95 TOTALS(L) = TOTALS(L) + 1                                       DSP01840
100 CONTINUE                                                         DSP01850
C*                                                                    DSP01860
C* HAVE 3 LINES BEEN READ                                         DSP01870
C*                                                                    DSP01880
IF(START)GO TO 105                                                 DSP01890
J=J+1                                                              DSP01900
IF(J.LT.3)GO TO 91                                                 DSP01910
START=.TRUE.                                                       DSP01920
GO TO 91                                                            DSP01930
C*                                                                    DSP01940
C* SPATIAL FILTERING                                              DSP01950
C*                                                                    DSP01960
105 IF(FILTER.EQ.0)GO TO 115                                       DSP01970
I=2                                                                DSP01980
106 IF(IR(I-1,12) .NE. IR(I+1,12)) GO TO 110                      DSP01990
IF(IR(I,11) .NE. IR(I,13)) GO TO 110                             DSP02000
IF(IR(I,11) .NE. IR(I-1,12)) GO TO 110                           DSP02010
IF(IR(I,12) .EQ. IR(I,11))GO TO 110                              DSP02020
IF(IR(I,11) .EQ. NOSUB3)GO TO 110                                 DSP02030
ICC=IR(I,11)                                                       DSP02040
ICK=IR(I,12)                                                       DSP02050
IF(ICK.EQ.NOSUB3)GO TO 110                                         DSP02060
TOTALS(ICC)=TOTALS(ICC)+1                                          DSP02070
TOTALS(ICK)=TOTALS(ICK)-1                                          DSP02080
SPKNT=SPKNT+1                                                       DSP02090
IR(I,12) = IR(I,11)                                                DSP02100
I=I+1                                                              DSP02110
110 IF(I.LE.PTS-1)GO TO 106                                         DSP02120
C*                                                                    DSP02130
C* GET PERFORMANCE FOR LINE II                                    DSP02140
C*                                                                    DSP02150
C* CODE ADDED NOV 13,1978 TO INCLUDE LIST PROCESSING             DSP02160
C*                                                                    DSP02170
C* IF (DOTKEY.EQ.0) GO TO 40                                       DSP02180
C* CONTINUE                                                         DSP02190
115                                                                    DSP02200
116                                                                    DSP02210
C* TEST TO SEE IF THE CURRENT LINE CONTAINS ANY DOTS             DSP02220
C*                                                                    DSP02230
IF (ILINE(I1).GT.TRNVER(2,NOFLD2)) GO TO 114                     DSP02240
BCNT = 0                                                            DSP02250
ECNT = 0                                                            DSP02260
                                                                    DSP02270
                                                                    DSP02280
```

ORIGINAL PAGE IS
OF POOR QUALITY

12-13

192

FILE DSPLY2

```

DO 41 I=1,NOFLD2
IF (ILINE(I1).NE.TRNVER(2,I)) GO TO 42
IF (RCNT.EQ.0) RCNT = I
GO TO 41
42 IF (RCNT.EQ.0) GO TO 41
ECNT = I-1
GO TO 43
41 CONTINUE
IF (ILINE(I1).NE.TRNVER(2,NOFLD2)) GO TO 114
ECNT = NOFLD2
C*
C* FOR DOT DATA PROCESSING, CALL THE INTERNAL SUBROUTINE DOTPCT TO
C* BUILD THE CLASSIFICATION PERFORMANCE TABLE (PCTABD)
C*
43 GO TO 432
40 IF (TSTKEY.EQ.0)CALL PCT(I1,IR(1,I1),TRNFLD,TRNVER,
* TRNSAV,PCTAB,NOFLD2,SAMSTR,SAMEND,SAMINC)
114 IF (TSTKEY.EQ.1)CALL PCT(I1,IR(1,I1),TSTFLD,ISTVER,
* TSTSAV,PCTAB,NOFLD3,SAMSTR,SAMEND,SAMINC)
IF (NOMAP.EQ.0) GO TO 135
C*
C*** CODE ADDED NOV 13,1978 TO INCLUDE LIST PROCESSING
C
C* IF (DOTKEY.NE.0) GO TO 117
C*
C* OUTLINE TRAINING AND/OR TEST FIELDS
C*
C* IF (TRNKEY.EQ.1)CALL FLDBOR(TRNNO,ILINE(I1),IR(1,I1),NOFLD2,
* TRNFLD,TRNSAV,TRNVER,NOSUB3,SAMSTR,SAMEND,
* SAMINC,LININC)
117 IF (TSTKEY.EQ.1)CALL FLDBOR(TSTNO,ILINE(I1),IR(1,I1),NOFLD3,
* TSTFLD,TSTSAV,TSTVER,NOSUB3,SAMSTR,SAMEND,
* SAMINC,LININC)
C* SET UP SYMBOLS FOR THIS LINE. FIRST MAKE SURE I/O FROM LAST LINE
C* IS COMPLETED.
DO 120 I=1,PTS
L = IR(I,I1)
IF (L.EQ.0)OUT(I)=BLANK
IF (L.NE.0)OUT(I)=SYMMTX(L)
120 CONTINUE
C*
C* WRITE FIRST 110 SAMPLES ON LINE PRINTER AND THE REST ON DRUM
C*
C* IPTS=PTS
C* IF (IPTS.GT.110)IPTS=110
C* IF (IPTS.LE.110)GO TO 125
C* IPD=PTS-110
C* IF (FULL)GO TO 125
C* CALL RWRITE(ADRES,OUT(111),IPD,LSTAT)
C* ADRES=ADRES+IPD
C* DRUMLN=DRUMLN+1
C* IF (ADRES+IPD.LE. DRUMAD+DRMWDS)GO TO 125
C* FULL=.TRUE.
125 WRITE(6,240) ILINE(I1),(OUT(I),I=1,IPTS)
C*
C* IS CLASSIFICATION MAP TO BE OUTPUT IN UNIVERSAL OR LARSYS
C* FORMAT
C*
135 IF (MAPFMT .LE. 0) GO TO 200
C*
C* CHECK TO SEE IF LAST WRITE IS COMPLETED
C*
C* GO TO (155,160), HDREC
C*
C* WRITE HEADER RECORD
C*
155 NC = 1
LINES = 0
FEAT = 1
LSTLIN = 0
HDREC = 2
NOFILE = NOFILE + 1
CALL WRTHED(NC,FEAT,PTS,MAPFMT,MAPUNT)
C*

```

DSP02290
 DSP02300
 DSP02310
 DSP02320
 DSP02330
 DSP02340
 DSP02350
 DSP02360
 DSP02370
 DSP02380
 DSP02390
 DSP02400
 DSP02410
 DSP02420
 DSP02430
 DSP02440
 DSP02450
 DSP02460
 DSP02470
 DSP02480
 DSP02490
 DSP02500
 DSP02510
 DSP02520
 DSP02530
 DSP02540
 DSP02550
 DSP02560
 DSP02570
 DSP02580
 DSP02590
 DSP02600
 DSP02610
 DSP02620
 DSP02630
 DSP02640
 DSP02650
 DSP02660
 DSP02670
 DSP02680
 DSP02690
 DSP02700
 DSP02710
 DSP02720
 DSP02730
 DSP02740
 DSP02750
 DSP02760
 DSP02770
 DSP02780
 DSP02790
 DSP02800
 DSP02810
 DSP02820
 DSP02830
 DSP02840
 DSP02850
 DSP02860
 DSP02870
 DSP02880
 DSP02890
 DSP02900
 DSP02910
 DSP02920
 DSP02930
 DSP02940
 DSP02950
 DSP02960
 DSP02970
 DSP02980
 DSP02990
 DSP03000
 DSP03010
 DSP03020
 DSP03030
 DSP03040

FILE DSPLY2

```

C*   WRITE DATA RECORD
C*
100  LINES = LINES + 1
    IF (LINES.EQ. LINES) LSTLIN = -1
    CALL WRTLN(IR(1,1),LSTLIN)
200  CONTINUE
    IF (ILINE(I3).EQ.0) GO TO 201
C*   SET INDICES AND GO READ NEXT LINE
C*
    J=1
    I1=1
    I2=1
    I3=J
    GO TO 91
C*
C*   LAST LINE IN THIS FIELD HAS BEEN READ. MAKE SURE LAST 2 LINES
C*   ARE PRINTED.
C*
201  LAST=LAST+1
    IF (LAST.EQ.2) GO TO 203
    I1=1
    I2=1
    GO TO 115
C*
C*   NOW FINISH PRINTING MAP FOR THIS FIELD.
C*
203  CONTINUE
    IF (MAPFMT.GT. 0) WRITE(6,2200)NOFILE,FLDESC,(FORMAT(I,MAPFMT),
    * I=1,3),LINES
2200  FORMAT(//T55,'FILE NO. - ',I6,/T55,'FIELD NAME - ',A4,/
    * T55,'FORMAT - ',I3A4,/T55,'NO. RECORDS - ',I6)
    IF (NOMAP.LE. 0) GO TO 230
C
    IF (PTS.LE.110) GO TO 230
    AD=0
    NBUFS=20
    IF (NBUFS.GT.LINES) NBUFS=LINES
    KPTS=PTS-110
    LPTS=110
    NWDTMS=KPTS/110
    IF (MOD(KPTS,110).NE.0) NWDTMS=NWDTMS+1
    LASTRC=MOD(KPTS,110)
    IF (LASTRC.EQ.0) LASTRC=110
    I=0
    I=I+1
219  IF (I.GT.NWDTMS) GO TO 220
    LINCNT=0
    WRTFG=2
    GO TO 370
221  PRTFG=2
    GO TO 510
222  CONTINUE
    IF (I.EQ.NWDTMS) LPTS=LASTRC
    ADRES=DRUMAD + AD
    DO 205 J=1,NBUFS
    CALL PREAD(ADRES,BUF(1,J),LPTS,JSTAT(J))
    ADRES=ADRES+KPTS
    LINCNT=LINCNT + 1
205  CONTINUE
    LINE=LINSTR
    IBUF=1
C*   FINISHED READING
210  IF (JSTAT(IBUF).EQ.1) GO TO 210
    WRITE(6,240)LINE,(BUF(IK,IBUF),IK=1,LPTS)
    LINE = LINE + LININC
    LINCNT=LINCNT+1
    IF (LINCNT.GT. DRUMLN) GO TO 215
    CALL PREAD(ADRES,BUF(1,IBUF),LPTS,JSTAT(IBUF))
    ADRES=ADRES + KPTS
215  IBUF=IBUF+1
    IF (IBUF.GT.NBUFS) IBUF=1
    IF (LINE.LE.LINEND) GO TO 210
    AD=AD + LPTS
    GO TO 219
220  CONTINUE

```

DSP03050
 DSP03060
 DSP03070
 DSP03080
 DSP03090
 DSP03100
 DSP03110
 DSP03120
 DSP03130
 DSP03140
 DSP03150
 DSP03160
 DSP03170
 DSP03180
 DSP03190
 DSP03200
 DSP03210
 DSP03220
 DSP03230
 DSP03240
 DSP03250
 DSP03260
 DSP03270
 DSP03280
 DSP03290
 DSP03300
 DSP03310
 DSP03320
 DSP03330
 DSP03340
 DSP03350
 DSP03360
 DSP03370
 DSP03380
 DSP03390
 DSP03400
 DSP03410
 DSP03420
 DSP03430
 DSP03440
 DSP03450
 DSP03460
 DSP03470
 DSP03480
 DSP03490
 DSP03500
 DSP03510
 DSP03520
 DSP03530
 DSP03540
 DSP03550
 DSP03560
 DSP03570
 DSP03580
 DSP03590
 DSP03600
 DSP03610
 DSP03620
 DSP03630
 DSP03640
 DSP03650
 DSP03660
 DSP03670
 DSP03680
 DSP03690
 DSP03700
 DSP03710
 DSP03720
 DSP03730
 DSP03740
 DSP03750
 DSP03760
 DSP03770
 DSP03780
 DSP03790
 DSP03800

FILE DSPLY2

```

240 FORMAT(' ',15,2X,110A1)
IF (FILTER.EQ.0) GO TO 230
WRITE (6,305) SPKNT
305 FORMAT('/ THE CLASSIFICATION OF ',I7,' PIXELS WAS CHANGED AS A RESU
*LT OF SPATIAL FILTERING' /)
230 CONTINUE
250 CONTINUE
C**
C** PRINT CLASSIFICATION SUMMARY FOR THIS FIELD
C**
C** CALL SETMRG(68,4,62)
C** CALL PRISUM(TOTALS,TTOL,FLDESC)
C** IF (DOTKEY.EQ.0) GO TO 500
C**
C** DOTS IN THE DESIGNATED AREA OR NOT IN THE CLASSIFIED AREA WILL
C** HAVE PCTABD=0
C*** CODE ADDED NOV 13 ,1978 TO INCLUDE LIST PROCESSING
C**
C** 255 CONTINUE
C**
C** WRITE (6,5)
C** 5 FORMAT ('H1)
C** DO 7 CHPCT=1,NOFLD2
C** IF (PCTABD(CHPCT,1).LE.NOSUB3) GO TO 8
C** WRITE (6,4) TRNVER(1,CHPCT),TRNVER(2,CHPCT)
C** 4 FORMAT (/3X, 'DOT (' , I4, ' , ' , I4, ' ) IS IN THE DESIGNATED AR
*EA' )
C** PCTABD(CHPCT,1) = 0
C** GO TO 7
C** 8 IF (PCTABD(CHPCT,1).GT.0) GO TO 7
C** WRITE (6,2) TRNVER(1,CHPCT),TRNVER(2,CHPCT)
C** 2 FORMAT (/3X, 'DOT (' , I4, ' , ' , I4, ' ) IS NOT IN THE CLASSIFI
*ED AREA' )
C** 7 CONTINUE
C** WRITE (6,5)
C*** CODE ADDED NOV 13, 1978 TO INCLUDE LIST PROCESSING
C**
C** FLDCNT = FLDCNT + 1
C** CALL LISTSM(TOTALS,TTOL,
* PCTABD,GTUNIT,GTFILE,AIUNIT,AIFILE,
* PPUNIT,PPFILE,NAMECT,ALP,FLDCNT,
* NOCAT,CATNAM,SUBCAT,NOFLD2,NOSUB2,SUBNAM)
C**
C** 500 CONTINUE
C** GO TO 20
C** 310 CONTINUE
C** RETURN
C**
C** SELF-CONTAINED SUBROUTINE TO PRINT HEADERS
C** -----
C** 370 WRITE (6,HEAD)
C** WRITE (6,380) FLDESC,COATE
C** 380 FORMAT (' DISPLAY OF CLASSIFIED FIELD.....',A4/
* CLASSIFICATION DATE.....',2A4)
C** IF (RMFLG.GT.0) WRITE (6,390) BMCOMB,(FETVC2(L),L=1,8MFEAT)
C** IF (RMFLG.EQ.0) WRITE (6,400) (FETVC2(L),L=1,NOFET2)
C** CALL MAPHD (NOCAT,SYMMTX,CATNAM,KATNU,CLSNAM,SUBNO,SUBNAM,
* CLSSUB,NOCLS2,NOSUB2,THRSKY,THRES)
C** IF (DFSKEY.EQ.1) WRITE (6,410) SYMMTX(DESUNI)
C** 410 FORMAT (/5X, 'DESIGNATED FIELDS SYMBOL IS ',A1/)
C** 390 FORMAT (12X, 'CLASSIFICATION CHANNELS...',I2, ' LINEAR COMBINATIONS
* CHANNELS' /3X, 30I3)
C** 400 FORMAT (12X, 'CLASSIFICATION CHANNELS...',30I3)
C** GO TO (31,221),WRTEG
C** -----
C**
C** INTERNAL ROUTINE TO PRINT COLUMN NUMBERS
C**
C** 510 J=0

```

FILE DSDLY2

```

DO 50 IJ=ISTRT,IEND,SAMINC
J=J+1
COL(1,J) = IJ/100
COL(2,J)=MOD(IJ,100)/10
COL(3,J)=MOD(IJ,10)
IF (J.EQ.110) GO TO 60
50 CONTINUE
60 SAMEN=IJ
ISTRT=SAMEN + SAMINC
IPTS=J
WRITE(6,96)
DO 80 IJ=1,3
80 WRITE(6,90) (COL(IJ,J),J=1,IPTS)
90 FORMAT(9X,11011)
WRITE(6,96)
96 FORMAT(/)
GO TO (85,222),PRTFG
*****
INTERNAL SUBROUTINE DOTPCT
*
* PURPOSE - TO SET PCTAB TO EACH DOT'S RESPECTIVE SUBCLASS
* (OR THRESHOLD) NUMBER WHEN THE DOTKEY FLAG IS ON
*****
*
* TEST FOR THE POSITION OF THE DOT IN THE CLASSIFICATION
* RECORD
*
* IF THE DOT'S POSITION IS NOT COMPATIBLE WITH THE CLASSIFICATION
* RECORD, PCTAB(JJ,1) = 0
432 IF (PCTKEY.NE.0) GO TO 430
DO 431 I=1,NOFLD2
431 PCTAB(I,1)=0
PCTKEY = 1
430 DO 411 K=HCNT,ECNT
SAMDF=TRNVER(1,K) + SAMSTR
SAMP5=SAMDF/SAMINC + 1
CHSP5=(SAMP5-1)*SAMINC
SAMDF = SAMDF + SAMSTR
CHSP5 = CHSP5 + SAMSTR
IF (CHSP5.NE.SAMDF) GO TO 421
PCTAB(K,1) = IR(SAMP5,11)
GO TO 411
421 PCTAB(K,1) = 0
411 CONTINUE
GO TO 114
C
END

```

DSP04570
 DSP04580
 DSP04590
 DSP04600
 DSP04610
 DSP04620
 DSP04630
 DSP04640
 DSP04650
 DSP04660
 DSP04670
 DSP04680
 DSP04690
 DSP04700
 DSP04710
 DSP04720
 DSP04730
 DSP04740
 DSP04750
 DSP04760
 DSP04770
 DSP04780
 DSP04790
 DSP04800
 DSP04810
 DSP04820
 DSP04830
 DSP04840
 DSP04850
 DSP04860
 DSP04870
 DSP04880
 DSP04890
 DSP04900
 DSP04910
 DSP04920
 DSP04930
 DSP04940
 DSP04950
 DSP04960
 DSP04970
 DSP04980
 DSP04990
 DSP05000
 DSP05010
 DSP05020
 DSP05030
 DSP05040
 DSP05050
 DSP05060
 DSP05070
 DSP05080

FILE: EMTHRS

```

SUBROUTINE EMTHRS(FLDSAV, FIELD, VERTEX, NOFLD)
-----
C THIS SUBROUTINE IS USED ONLY WHEN THE EMPIRICAL THRESHOLD OPTION
C REQUESTED
C OR IF HISTOGRAMS OF THE QUADRATIC FORM WERE REQUESTED WITH SOME
C OTHER THRESHOLDS
-----
C IMPLICIT INTEGER (A-Z)
C REAL VR, DSFUNC, THRES, CON
C INCLUDE CMHKK6.LIST
C COMMON/CMHKK10.LIST
COMMON/GLOBAL/HEAD(63), MAPTAP, DATAPE, SAVTAP, RMFILE, RMKEY,
* HISFIL, HISKEY, TRFORM, ERPTP, ERPKEY, MAPUNT, NOFILE,
* DRUMAD, DRUMWDS, PAGESIZ, DATFIL, STAFIL, ASAV, ASAVFL
* NMSTUN, NMSTFI, SCTRUN, MAPFIL
* DOTUNT, DOTFIL, NCHPAS, TRNSFL, BMTRFL, HISTFL, PCHUNT,
* CRDUNT, PHTUNT, RANDIO
COMMON/DISPL/CATFLG, CATNAM(61), CLSNAM(61), SURNAM(61), SURN0(60),
* SUBCAT(60), CLSSUB(60), NOMAP, TOTVT3, NOSUR3,
* PCFKEY, ISTKEY, TRNKEY, THRSKY, STATKY, EMPTRS, THRSVA,
* PLTKEY, RMFLG, RMCOMB, RMFEAT, CDATE(2),
* FLDSV2, FIELD2, VERTX2, FLDSV3, FIELD3, VERTX3, PCTID3,
* THRS(60), SYMMTX(66), HIGH(60), CON(60)
* FLDKEY, NOFLD2, NOFLD3, NOFET2, FETVC2(30)
* NOSUR2, NOTREFD, TOTVT2, NOCLS2
* KATNO(60), NOCAT, FILTER, MAPFMT
* DESKEY, DESUNI, DESOth, CROP, ACROP, AOTHER, ATOTAL
* SITE(6), ANALYS(5), CAM(15), CRPKEY, KEPPTS(60)
* DOTKEY, DOTERR
C$END
DATA RANGE/100/
DIMENSION DSFUNC(100,60), IR(1000), VR(1000), FLDINF(6), SCRAT(500)
EQUIVALENCE (SCRAT, IR)
10 READ(MAPTAP) FLDINF, PTS, LINES, FLDDESC
IF (PTS.EQ.0) GO TO 40
DO 30 J=1, LINES
READ(MAPTAP) ILINE, (IR(J), J=1, PTS), (VR(J), J=1, PTS)
C* HISTOGRAM VR WITHIN GROUND TRUTH FIELDS
CALL PCTT(ILINE, IR, VR, FIELD, VERTEX, FLDSAV, DSFUNC, NOFLD,
* FLDINF(4), FLDINF(5), FLDINF(6), CON, RANGE)
30 CONTINUE
C* READ END OF FIELD RECORD AND GO SEE IF THERE IS ANOTHER FIELD
READ(MAPTAP) ILINE
GO TO 10
C*
C* ALL FIELDS ON THIS FILE HAVE BEEN PROCESSED NOW PLOT THE HISTOGRAM
C*
40 CONTINUE
CALL DISTCV(DSFUNC, SCRAT, RANGE)
C*
C* NOW GO BACK TO BEGINNING OF THIS FILE AND POSITION TAPE OVER THE
C* FOUR HEADER RECORDS - GETTING IT READY DSPLY2
C*
REWIND MAPTAP
CALL F94SFL(MAPTAP, 4, ISTAT)
IF (ISTAT.GT. 0) WRITE(6,500) ISTAT
500 FORMAT(' ERROR BACK SPACING MAPTAP', ISTAT = ', IS)
IF (ISTAT.GT. 0) CALL CMERR
DO 210 J=1, 4
210 READ(MAPTAP)
RETURN
END

```

FILE: FDIST

```

SUBROUTINE FDIST
ROUTINE TO USE FISHIN TO GET THRESHOLD VALUES
DIMENSION F(60)
INCLUDE CMRK10.LIST
COMMON/ISPL/CATFLG,CATNAM(61),CLSNAM(61),SURNAM(61),SUBNO(60),
* SUBCAT(60),CLSSUB(60),NOMAP,TOTVT3,NOSUB3,
* PCFKEY,TSTKEY,TRNKEY,THRSKY,STATKY,EMPTRS,THRSVA,
* PLTKEY,BMFLG,BMCOMB,BMFEAT,CDATE(2),
* FLDV2,FLD2,VERTX2,FLDSV3,FLD3,VERTX3,PCTID3,
* THRS(60),SYMMTX(66),HIGH(60),CON(60)
* FLDKEY,NOFLD2,NOFLD3,NOFET2,FETVC2(30)
* NOSUB2,NOTHFD,TOTVT2,NOCLS2
* KATNO(60),NOCAT,FILTER,MAPFMT
* DESKEY,DESUNI,DESOTH,CROP,ACROP,AOTHR,ATOTAL
* SITE(6),ANALYS(5),CAM(15),CRPKEY,KEPPTS(60)
* DOTKEY,DOTERR
C&END
NOSUB2 = NUMBER OF SUBCLASSES
THRS(I) CONTAINS INPUT CONFIDENCE LEVELS
DO 10 I=1,NOSUB2
10 F(I)=1-THRS(I)
NX = NUMBER OF SAMPLES
NOFET2 = NUMBER OF CHANNELS
COMPUTE THRESHOLD VALUES USING FISHER F-DISTRIBUTION FUNCTION
DO 20 I=1,NOSUB2
FX = KEPPTS(I)
NS=KEPPTS(I)-NOFET2
C- IFLAG=0
VAP=FISHIN(F(I),NOFET2,NS,IFLAG)
IF(IFLAG.EQ.1) GOTO 15
C- FK=(NOFET2*(FX-1)*(FX+1))/(NS*FX)
C- THRS(I) = FK*VAR
C- GO TO 20
C-
C- THRESHOLD WILL BE SET TO 999.999 IF OVERFLOW OCCURS IN FISHIN.
15 THRS(I)=999.999
WRITE(6,1)I
11 FORMAT(AX,'FDIST- OVERFLOW CONDITION IN FISHIN ROUTINE FOR SUBCLASS
15=.14,.1. THRESHOLD SET TO 999.999')
C-
C- 20 CONTINUE
C- RETURN
END

```

FDI00010
FDI00020
FDI00030
FDI00040
FDI00050
FDI00060
FDI00070
CMB00010
CMB00020
CMB00030
CMB00040
CMB00050
CMB00060
CMB00070
CMB00080
CMB00090
CMB00100
CMB00110
CMB00120
FDI00090
FDI00100
FDI00110
FDI00120
FDI00130
FDI00140
FDI00150
FDI00160
FDI00170
FDI00180
FDI00190
FDI00200
FDI00210
FDI00220
FDI00230
FDI00240
FDI00250
FDI00260
FDI00270
FDI00280
FDI00290
FDI00300
FDI00310
FDI00320
FDI00330
FDI00340
FDI00350
FDI00360
FDI00370
FDI00380
FDI00390
FDI00400
FDI00410
FDI00420
FDI00430
FDI00440
FDI00450
FDI00460
FDI00470
FDI00480
FDI00490
FDI00500

FILE: FISH

```

FUNCTION FISH(F,N1,N2)
LOGICAL E1,E2,E3
E1=.FALSE.
E2=.FALSE.
E3=.FALSE.
IF (MOD(N1,2).EQ.0) E1=.TRUE.
IF (MOD(N2,2).EQ.0) E2=.TRUE.
X=N2/(N2+N1*F)
IF (.NOT.(E1.OR.E2)) GO TO 5
IF (E1.AND..NOT.E2) GO TO 1
IF (.NOT.E1.AND.E2) GO TO 2
IF (N1.LE.N2) GO TO 1
2  I=N1
   N1=N2
   N2=I
   X=1.0-X
   E3=.TRUE.
1  Y=1.0-X
   FISH=0.0
   H=SQRT(X**N2)
   M=N1/2-1
   I=-1
3  I=I+1
   IF (I.GT.M) GO TO 10
   FISH=FISH+H
   H=(H*Y*(N2+2.*I))/(2.*(I+1.))
   GO TO 3
10 IF (E3) GO TO 4
   FISH=1.0-FISH
   RETURN
4  I=N1
   N1=N2
   N2=I
   RETURN
5  Y=1.0-X
   H=.63661977*SQRT(X*Y)
   FISH=.63661977*ARCOS(SQRT(X))
   IF (N2.EQ.1) GO TO 6
   M=N2-2
   DO 4 I=1,M,2
     FISH=FISH+H
     H=H*X*(I+1)/(I+2)
6  IF (N1.EQ.1) RETURN
     H=H*N2
     M=N1-2
     DO 7 I=1,M,2
       FISH=FISH-H
       H=H*Y*(N2+I)/(I+2)
7  RETURN
END

```

```

FIS000010
FIS000020
FIS000030
FIS000040
FIS000050
FIS000060
FIS000070
FIS000080
FIS000090
FIS000100
FIS000110
FIS000120
FIS000130
FIS000140
FIS000150
FIS000160
FIS000170
FIS000180
FIS000190
FIS000200
FIS000210
FIS000220
FIS000230
FIS000240
FIS000250
FIS000260
FIS000270
FIS000280
FIS000290
FIS000300
FIS000310
FIS000320
FIS000330
FIS000340
FIS000350
FIS000360
FIS000370
FIS000380
FIS000390
FIS000400
FIS000410
FIS000420
FIS000430
FIS000440
FIS000450
FIS000460
FIS000470
FIS000480
FIS000490
FIS000500

```

FILE: FISHIN

```

FUNCTION FISHIN(ALPHA,N1,N2,IFLAG)
Y1=N1
Y2=N2
IF(N1.EQ.1) Y1=2
IF(N2.EQ.1) Y2=2
X=TINORM(1.-ALPHA,IFLAG)
IF(IFLAG.EQ.1) GOTO 6
Y=(X**2-3.)/6.
IC=0
Y1=1./(Y1-1.)
Y2=1./(Y2-1.)
H=2./(Y1+Y2)
X=X*SQRT(H*Y)/H-(Y1-Y2)*(Y+5./6.-2./(3.*H))
X=EXP(2.*X)
G=1.
IR1=2
IF(MOD(N1,2).EQ.0) GO TO 1
G=1.7724539
IR1=1
1 IR2=2
IF(MOD(N2,2).EQ.0) GO TO 2
G=G*1.7724539
IR2=1
2 IR3=2
IF(MOD(N1+N2,2).EQ.0) GO TO 3
G=G*1.7724539
IR3=1
3 IF((IR1+IR2).NE.2) G=2.*G
IF((N1+N2).LE.3) GO TO 5
ND=N1+N2-2-IR3
I=-2
I=I+2
IF(I.GT.ND) GO TO 5
IF((IR1+1).LE.(N1-2)) G=G*(IR1+1)
IF((IR2+1).LE.(N2-2)) G=G*(IR2+1)
G=G/(IR3+1)
GO TO 4
4 Y2=N2/(N2*N1*X)
Y1=1.-Y2
Y=1.+(G*(1.-ALPHA-FISH(X,N1,N2)))/SQRT(Y1*N1*Y2*N2)
FISHIN=X*Y
IF(Y.LT.0.) FISHIN=.5*X
IF(ABS(X/FISHIN-1.).LT.(.5E-6)) GO TO 7
IF(ABS(X-FISHIN).LT.(.5E-6)) GO TO 7
IC=IC+1
IF(IC.GT.100) RETURN
X=FISHIN
GO TO 6
6 IFLAG=1
7 RETURN
END

```

```

F 500010
F 500020
F 500030
F 500040
F 500050
F 500060
F 500070
F 500080
F 500090
F 500100
F 500110
F 500120
F 500130
F 500140
F 500150
F 500160
F 500170
F 500180
F 500190
F 500200
F 500210
F 500220
F 500230
F 500240
F 500250
F 500260
F 500270
F 500280
F 500290
F 500300
F 500310
F 500320
F 500330
F 500340
F 500350
F 500360
F 500370
F 500380
F 500390
F 500400
F 500410
F 500420
F 500430
F 500440
F 500450
F 500460
F 500470
F 500480
F 500490
F 500500
F 500510

```


FILE: FLOBOR

```

      SURROUTINE FLDPOW (ISYM, LINUM, IR, NOFLD, FIELD, FLOSAV, VERTEX,
      NOSUB3, SAMSTR, SAMEND, SAMINC, LININC)
C*
C* THIS SURROUTINE SETS THE SYMBOL INDEX IN THE CLASSIFIED LINE (IR)
C* ARRAY TO OUTLINE TRAINING OR TEST FIELDS IN THE MAP.
C*
      IMPLICIT INTEGER (A-Z)
      DIMENSION IN(1), FIELD(5, NOFLD), FLOSAV(4, NOFLD), VERTEX(1)
      DIMENSION FL(22)
      DO 50 I=1, NOFLD
      IF (LINUM+LININC .LT. FIELD(1, I)) GO TO 50
      IF (LINUM-LININC .GT. FIELD(2, I)) GO TO 50
      IF (FIELD(3, I) .GT. SAMEND) GO TO 50
      IF (FIELD(4, I) .LT. SAMSTR) GO TO 50
C*
C* FOUND A FIELD THAT NEEDS BORDER ON THIS LINE. NOW FIND FIELD
C* INTERSECTIONS ON THIS LINE.
C*
      NV=FLOSAV(4, I)
      IPT=FIELD(5, I)
C*
C* TOP OR BOTTOM
C*
      IF (LINUM.GT.FIELD(2, I)) GO TO 40
      IF (LINUM.LT.FIELD(1, I)) GO TO 30
      CALL FDLINT (VERTEX(IPT), NV, FL, LINUM, SAMPS, NI)
      DO 20 J=1, NI-2
      IR = (FL(J)-SAMSTR)/SAMINC
      IE = (FL(J+1)-SAMSTR)/SAMINC + 2
      IF (MOD(SAMSTR, SAMINC).NE.MOD(FL(J), SAMINC)) IB=IB+1
      RORNUM=ISYM
      IF (IR(IH).GT.NOSUB3) RORNUM = NOSUB3 + 3
      IR(IR)=RORNUM
      IF (IR(IF).GT.NOSUB3) RORNUM = NOSUB3 + 3
      20 IR(IF)=RORNUM
      GO TO 50
      30 CONTINUE
C*
C* GET INTERCEPTS FOR TOP LINE IN FIELD
C*
      CALL FDLINT (VERTEX(IPT), NV, FL, FIELD(1, I), SAMPS, NI)
      GO TO 45
C*
C* GET INTERCEPTS FOR BOTTOM LINE IN FIELD
C*
      40 CALL FDLINT (VERTEX(IPT), NV, FL, FIELD(2, I), SAMPS, NI)
      45 DO 47 J=1, NI-2
      IB = (FL(J)-SAMSTR)/SAMINC
      IE = (FL(J+1)-SAMSTR)/SAMINC + 2
      IF (MOD(SAMSTR, SAMINC).NE.MOD(FL(J), SAMINC)) IB=IB+1
      DO 46 IJ=IH, IE
      RORNUM=ISYM
      IF (IR(IJ).GT.NOSUB3) RORNUM=NOSUB3+3
      46 IR(IJ)=RORNUM
      47 CONTINUE
      50 CONTINUE
      RETURN
      END

```

FLOR0720

FILE: LISTPR

```

C THIS SUBROUTINE PRINTS LABEL TABLES
SUBROUTINE LISTPR(ISIT,DOTLAB,ITYPE,SUBLAB)
IMPLICIT INTEGER (A-Z)
DATA SLASH/'/' , BLANK/' '
DIMENSION LINE(57),SUBLAB(19,11),LINE2(19)
      DIMENSION DOTLAB(19,11,4)
      I1 = ISIT
      I2 = 4
      NPRT = 6
      WRITE(NPRT,10)
10    FORMAT(1H)
      IF (ISIT.EQ.1) WRITE(NPRT,120)
120   FORMAT(//,50X,'APTC VS CLASSIFIED LABELS ')
      IF (ISIT.EQ.2) WRITE(NPRT,130)
130   FORMAT(//,50X,'GROUND TRUTH VS CLASSIFIED LABELS')
      IF (ISIT.EQ.3) WRITE(NPRT,230)
230   FORMAT(//,50X,'A. I. LABELS VS CLASSIFIED LABELS')
      WRITE(NPRT,20) ITYPE
20    FORMAT(//,50X,'TYPE ',I1,' DOT CLASSIFICATION')
      WRITE(NPRT,30) (I,I=10,190,10)
30    FORMAT(//,12X,19(15,1X))
      DO 200 J=1,11
        DO 215 J = 1,57
          LINE(J) = BLANK
215   DO 216 J=1,19
          LINE2(J)=BLANK
216   DO 220 J=1,19
          LL=3*(J-1)
          LINE(LL+1)=DOTLAB(J,I,11)
          LINE(LL+3)=DOTLAB(J,I,12)
          IF (DOTLAB(J,I,11).NE.BLANK) LINE(LL+2) = SLASH
          LL2 = LINE(LL+2)
          IF (LL2.NF.SLASH) LINE(LL+3) = BLANK
          IF (LL2.NF.SLASH) GO TO 220
          LINE2(J)=SUBLAB(J,I)
220   CONTINUE
          I10=1*10
          WRITE(NPRT,300) I10,(LINE(KK),KK=1,57),(LINE2(KK),KK=1,19)
300   FORMAT(//,9X,13,19(1X,1A2,1A1,1A2),/,12X,19(1X,1A4,1X))
200   CONTINUE
100   CONTINUE
      RETURN
      END

```

500010
 500020
 500030
 500040
 500050
 500060
 500070
 500080
 500090
 500100
 500110
 500120
 500130
 500140
 500150
 500160
 500170
 500180
 500190
 500200
 500210
 500220
 500230
 500240
 500250
 500260
 500270
 500280
 500290
 500300
 500310
 500320
 500330
 500340
 500350
 500360
 500370
 500380
 500390
 500400
 500410
 500420
 500430

FILE LISTSM

```

      SUBROUTINE LISTSM(TOTALS,TTOL,PCTAB,GTUNIT,
      * GTFILE,AIUNIT,AIFILE,PPUNIT,PPFILE,NAMECT,ALP,
      * FLDINT,NCAT,CATNM,SUBCAT,NFLD2,NSUB2,SUBNAM)
      IMPLICIT INTEGER (A-Z)

C*****
C*****      SUBROUTINE WRITTEN NOV 1978 TO INCLUDE LIST PROCESSING
C*****      M AND R ARE BOUNDARY DOT NAMES
C*****      D IS THE DESIGNATED NAME
C*****      NAMECT IS THE NAME OF THE 'SMALL GRAINS' CATEGORY
C*****      THE CATEGORY CLASSIFIER MUST HAVE BEEN INVOKED
C
      COMMON /LISTMM/ NPGA(3,2),NAMPGA(209,3,2),LINPGA(209,3,2),
      * SAMPGA(209,3,2),DOTLAB(209,4,2),VPGA(3),IPGA
C2345678
      INCLUDE CMBK14
      INCLUDE COMRK1
      COMMON/INFORM/NOCLS2,NOSUB2,NOFET2,VARS2,TOTVT2,NOFLD2,
      * AVER2,COVAR2,CLSID2,SURNO2,SURDS2,FLOSV2,VERTX2,
      * FETVC2(30),SUPVC2(75),SUHPTH(75),CLSV2(60),
      * KFPPTS(60),NOGRP,GRPNAM(60),GRPDEX(61),
      * GRPCHK(61),GROUPS(124)
      COMMON/DOTVEC/TYPE,CATNAM(60),NOCAT,TOTVEC,FLDINF(6),
      * PRTKEY,SIZE,LACIE
CSEND
      DIMENSION ALPMMSG(3),SURNAM(1),SUBLAB(209)
      DATA SYMTHR/1,1,1,SYNDES/1,1,1,SYMOUT/1,1,1,
      * MRCD/1,1,1,RRCD/1,1,1,DHCD/1,1,1,BLANK/1,1,1,
      * ALPMMSG/PPC 1,GT 1,AI 1/

C
      DIMENSION TTOL(1),PCTAB(1),CATNM(1),SUBCAT(1),
      * FIELDS(4,250),VERTEX(1000),INFUNT(3),INFFIL(3)

C
      REAL TOTALS(1),ALP(2)

C
      CODE ADDED TO PRINT DOT PERFORMANCE SUMMARY

      REAL RS,VRS,RC,VRC,TRM1,TRM3,ALPSUM,PG,CLP
      REAL SUM,RASE,DUTTOT(60),LABTOT(60),TOTCAT(60)
      REAL ALPHA(15),TRM2

C
      THIS CODE ADDED TO PRINT CONFUSION MATRIX N*H

      DIMENSION CONF(60,60),DOTNUM(209,2)

      REAL PHACH(2),PIXTOT,P11,P12,PB1,PB2,P,CATTOT(61)
      CONTINUE
      IF (FLDINT.GT.1) GO TO 400

C*****
C*****      INITIALIZE IF FIRST FIELD TO BE SUMMARIZED
C
      DO 10 I = 1,3
      DO 10 II = 1,2
      10  NPGA(I,II) = 0
      DO 20 I = 1,2
      DO 20 II = 1,3
      DO 20 III = 1,209
      NAMPGA(III,II,I) = BLANK
      DOTLAB(III,II,I) = BLANK
      20  CONTINUE

C
      READ IN PPC GT AI FILES
      ASSUME TYPE 1 AND 2 ON SAME UNIT    BACK-TO-BACK
C
      IPGA = 0
      IF (PPUNIT.EQ.0) GO TO 25
      IPGA = IPGA + 1
      VPGA(IPGA) = 1
      25  IF (GTUNIT.EQ.0) GO TO 30
      IPGA = IPGA + 1
      VPGA(IPGA) = 2
      30  IF (AIUNIT.EQ.0) GO TO 35
      IPGA = IPGA + 1
      VPGA(IPGA) = 3

```

```

S00010
S00020
S00030
S00040
S00050
S00060
S00070
S00080
S00090
S00100
S00110
S00120
S00130
S00140
S00150
S00160
S00170
S00180
S00190
S00200
S00210
S00220
S00230
S00240
S00250
S00260
S00270
S00280
S00290
S00300
S00310
S00320
S00330
S00340
S00350
S00360
S00370
S00380
S00390
S00400
S00410
S00420
S00430
S00440
S00450
S00460
S00470
S00480
S00490
S00500
S00510
S00520
S00530
S00540
S00550
S00560
S00570
S00580
S00590
S00600
S00610
S00620
S00630
S00640
S00650
S00660
S00670
S00680
S00690
S00700
S00710
S00720
S00730
S00740
S00750
S00760

```

FILE LISTSM

35 IF (IPGA.EQ.0) CALL CMERR

```
INFUNT(1) = PPUNIT
INFFIL(1) = PPFILE - 1
INFUNT(2) = GTUNIT
INFFIL(2) = GTFILE - 1
INFUNT(3) = AIUNIT
INFFIL(3) = AIFILE - 1
```

PCTAB STORED IN ORDER OF FIRST LINE---SECOND LINE ETC.

```
400 NSUB3 = NSUB2 + 1
    NSUB6 = NSUB2 + 5
    NSUB7 = NSUB2 + 6
    DO 401 I = 1,NFLO2
    SUBCL = PCTAB(I)
    IF (SURCL.GT.0) GO TO 405
    DOTLAB(I,4,1) = SYMOUT
    GO TO 420
405 IF (SURCL.LE.NSUB2) GO TO 410
    IF (SURCL.EQ.NSUB3) DOTLAB(I,4,1) = SYMTHR
    IF (SURCL.EQ.NSUB6.OR.SUBCL.EQ.NSUB7) DOTLAB(I,4,1) = SYMDES
    GO TO 420
410 CAT = SURCAT(SURCL)
    DOTLAB(I,4,1) = CATNM(CAT)
420 DOTLAB(I,4,2) = DOTLAB(I,4,1)
    SURLAB(I) = SURNAM(SURCL)
    IF (SURCL.EQ.NSUB6.OR.SURCL.EQ.NSUB7) SURLAB(I) = SYMDES
430 CONTINUE
```

COMPUTE TOTAL NUMBER OF CLASSIFIED DOTS

```
PIXTOT = 0.
DO 440 I = 1,NSUB2
440 PIXTOT = PIXTOT + TOTALS(I)
```

COMPUTE TOTAL NO. OF PIXELS IN EACH MACHINE CATEGORY
COUNT UP PIXELS OF CHOSEN AND OTHER CATEGORIES

```
PMACH(1) = 0.
PMACH(2) = 0.
DO 445 I = 1,61
445 CATTOT(I) = 0.
    DO 450 J = 1,NSUB2
    CAT = SURCAT(I)
    CATTOT(CAT) = CATTOT(CAT) + TOTALS(I)
    DUM = CATNM(CAT)
    IF (DUM.NE.NAMECT) GO TO 448
    ICAT = CAT
    PMACH(1) = PMACH(1) + CATTOT(CAT)
    GO TO 450
448 PMACH(2) = PMACH(2) + CATTOT(CAT)
450 CONTINUE
```

C23456789 NCAT = NO. OF MACHINE CLASSIFIED CATEGORIES
C23456789 ICAT = CATEGORY NUMBER OF PREFERRED CATEGORY

```
PMACH(1) = PMACH(1)/PIXTOT
PMACH(2) = PMACH(2)/PIXTOT
```

MAJOR LOOP

```
DO 600 I = 1,IPGA
```

CODE ADDED TO PRINT CONFUSION MATRIX N=0

```
DO 15 IL=1,60
DOTTOT(IL)=0
15 CONTINUE
    SUM=0
    DO 22 IJ=1,2
    DO 22 IK=1,209
    DOTNUM(IK,IJ)=0
```

500770
500780
500790
500800
500810
500820
500830
500840
500850
500860
500870
500880
500890
500900
500910
500920
500930
500940
500950
500960
500970
500980
500990
501000
501010
501020
501030
501040
501050
501060
501070
501080
501090
501100
501110
501120
501130
501140
501150
501160
501170
501180
501190
501200
501210
501220
501230
501240
501250
501260
501270
501280
501290
501300
501310
501320
501330
501340
501350
501360
501370
501380
501390
501400
501410
501420
501430
501440
501450
501460
501470
501480
501490
501500
501510
501520

FILE LISTSM

```

22  CONTINUE
    ISIT = VPGA(I)
    IPT = 0
    STAMNT = 1
    TYPE = 1
    NOCAT = 0
    NOFLD2 = 0
    TOTVT2 = 0
    TOTVEC = 0
    SWCHG = 0
    INIT = 0
110  CALL LISTLC(FIELDS,STAMNT,&130,&140,&150,SWCHG,
    *  INIT,INFUNT(ISIT),INFFIL(ISIT),IPT,VERTEX)
C
C 130  NPGA(ISIT,TYPE) = NOFLD2
    NAMPGA(NOFLD2,ISIT,TYPE) = FIELDS(1,NOFLD2)
    LINPGA(NOFLD2,ISIT,TYPE) = FLDINF(1)
    SAMPGA(NOFLD2,ISIT,TYPE) = FLDINF(4)
    I10 = FLDINF(1)/10
    I110 = FLDINF(4)/10
    J = (I10 - 1)*19 + I110
    DOTLAB(J,ISIT,TYPE) = FIELDS(1,NOFLD2)
C
C  CODE ADDED TO PRINT CONFUSION MATRIX N*N
C
IF (TYPE.EQ.1) GO TO 110
DO 135 JJ=1,NOCAT
IF (DOTLAB(J,ISIT,TYPE).EQ.CATNAM(JJ)) DOTNUM(J,1)=JJ
135  CONTINUE
    IF (DOTLAB(J,ISIT,TYPE).EQ.BLANK) GO TO 110
    SUM=SUM+1
    IF (DOTNUM(J,1).EQ.0) DOTNUM(J,1)=NOCAT+1
    GO TO 110
C
C***  DOT TYPE CHANGE
C
140  NOFLD2 = 0
    NOCAT=0
    GO TO 110
C
C***  SEND CARD IMAGE DETECTED
C
150  CONTINUE
C
C  CODE ADDED TO FIND CATEGORY TOTALS FOR DOT REPORT
C
DO 155 JJ=1,NOCAT
TOTCAT(JJ)=0
DO 155 JK=1,NOCAT
IF (CATNAM(JJ).EQ.CATNM(JK)) TOTCAT(JJ)=CATTOT(JK)
155  CONTINUE
C
C  CODE ADDED TO FIND DOT LABEL CATEGORY NUMBER
C
DO 160 JJ=1,209
IF (PCTAB(JJ).EQ.NSUB3) GO TO 160
IF (PCTAB(JJ).EQ.NSUB6) GO TO 160
IF (PCTAB(JJ).EQ.NSUB7) GO TO 160
DO 160 KK=1,NOCAT
IF (DOTLAB(JJ,4,2).EQ.CATNAM(KK)) DOTNUM(JJ,2)=KK
160  CONTINUE
IF (NPGA(ISIT,1).EQ.0) GO TO 505
ITYPE = 1
CALL LISTPR(ISIT,DOTLAB,ITYPE,SUPLAB)
505  IF (NPGA(ISIT,2).EQ.0) GO TO 600
ITYPE = 2
CALL LISTPR(ISIT,DOTLAB(1,1,2),ITYPE,SUPLAB)
C
C*** COMPUTE N11,N12,N21,N22,NR1,NR2 FOR TYPE 2 DOTS
C
N11 = 0
N12 = 0
N21 = 0
N22 = 0
NR1 = 0

```

LIS01530
 LIS01540
 LIS01550
 LIS01560
 LIS01570
 LIS01580
 LIS01590
 LIS01600
 LIS01610
 LIS01620
 LIS01630
 LIS01640
 LIS01650
 LIS01660
 LIS01670
 LIS01680
 LIS01690
 LIS01700
 LIS01710
 LIS01720
 LIS01730
 LIS01740
 LIS01750
 LIS01760
 LIS01770
 LIS01780
 LIS01790
 LIS01800
 LIS01810
 LIS01820
 LIS01830
 LIS01840
 LIS01850
 LIS01860
 LIS01870
 LIS01880
 LIS01890
 LIS01900
 LIS01910
 LIS01920
 LIS01930
 LIS01940
 LIS01950
 LIS01960
 LIS01970
 LIS01980
 LIS01990
 LIS02000
 LIS02010
 LIS02020
 LIS02030
 LIS02040
 LIS02050
 LIS02060
 LIS02070
 LIS02080
 LIS02090
 LIS02100
 LIS02110
 LIS02120
 LIS02130
 LIS02140
 LIS02150
 LIS02160
 LIS02170
 LIS02180
 LIS02190
 LIS02200
 LIS02210
 LIS02220
 LIS02230
 LIS02240
 LIS02250
 LIS02260
 LIS02270
 LIS02280

ORIGINAL PAGE
OF POOR QUALITY

FILE LISTSM

```

      NB2 = 0
C
C      CODE ADDED TO PRINT CONFUSION MATRIX N*N
      DO 510 IJ=1,60
      DO 510 IK=1,60
      CONF(IK,IJ)=0
510    CONTINUE
      DO 512 IJ=1,60
      DOTTOT(IJ)=0
      LARTOT(IJ)=0
512    CONTINUE
C
C      IF DOT PROCESSING SKIP LIST REPORTS
C
C      IF (NAMECT.EQ.BLANK) GO TO 582
C
      DO 580 II = 1,209
      DUM = SAMPGA(II,ISIT,2)
      IF (DUM.NE.NAMECT) GO TO 530
C*** DOT LABEL IS PREFERRED CATEGORY
C
      DUMS = SAMPGA(II,ISIT,2)/10
      DURL = LAMPGA(II,ISIT,2)/10
      DO 515 III = 1,11
      DO 514 IIII = 1,19
      IF (IIII.NE.DUMS) GO TO 514
      IF (IIII.NE.DURL) GO TO 514
C*** FOUND MACHINE CLASSIFIED DOT
      J = (III - 1)*19 + IIII
      DUMA = DOTLAR(J,4,2)
      IF (DUMA.EQ.DUM) N11 = N11 + 1
      IF (DUMA.EQ.SYMDES) GO TO 514
C23456789
      IF (DUMA.EQ.SYMTHR) GO TO 514
      IF (DUMA.EQ.SYMOUT) GO TO 514
      IF (DUM.NE.DUMA) N12 = N12 + 1
514    CONTINUE
515    CONTINUE
      GO TO 580
C
C***
C
C      530 IF (DUM.NE.MBCD.AND.DUM.NE.RBCD) GO TO 550
C*** THIS PIXEL WAS LABELED BOUNDARY
C
      DUMS = SAMPGA(II,ISIT,2)/10
      DURL = LAMPGA(II,ISIT,2)/10
      DO 535 III = 1,11
      DO 534 IIII = 1,19
      IF (IIII.NE.DUMS) GO TO 534
      IF (IIII.NE.DURL) GO TO 534
      J = (III - 1)*19 + IIII
      DUMA = DOTLAR(J,4,2)
      IF (DUMA.EQ.NAMECT) N11 = N11 + 1
      IF (DUMA.EQ.SYMDES) GO TO 534
      IF (DUMA.EQ.SYMTHR) GO TO 534
      IF (DUMA.EQ.SYMOUT) GO TO 534
      IF (DUMA.NE.NAMECT) N12 = N12 + 1
534    CONTINUE
535    CONTINUE
      GO TO 580
C
C*** THIS PIXEL IS LABELED DESIGNATED OR IS IN THE OTHER CATEGORY
C
550 IF (DUM.EQ.DHCD) GO TO 580
C*** IT'S IN THE OTHER CATEGORY
      DUMS = SAMPGA(II,ISIT,2)/10
      DURL = LAMPGA(II,ISIT,2)/10
      DO 555 III = 1,11
      DO 554 IIII = 1,19
      IF (IIII.NE.DUMS) GO TO 554
      IF (IIII.NE.DURL) GO TO 554
      J = (III - 1)*19 + IIII

```

LIS02290
 LIS02300
 LIS02310
 LIS02320
 LIS02330
 LIS02340
 LIS02350
 LIS02360
 LIS02370
 LIS02380
 LIS02390
 LIS02400
 LIS02410
 LIS02420
 LIS02430
 LIS02440
 LIS02450
 LIS02460
 LIS02470
 LIS02480
 LIS02490
 LIS02500
 LIS02510
 LIS02520
 LIS02530
 LIS02540
 LIS02550
 LIS02560
 LIS02570
 LIS02580
 LIS02590
 LIS02600
 LIS02610
 LIS02620
 LIS02630
 LIS02640
 LIS02650
 LIS02660
 LIS02670
 LIS02680
 LIS02690
 LIS02700
 LIS02710
 LIS02720
 LIS02730
 LIS02740
 LIS02750
 LIS02760
 LIS02770
 LIS02780
 LIS02790
 LIS02800
 LIS02810
 LIS02820
 LIS02830
 LIS02840
 LIS02850
 LIS02860
 LIS02870
 LIS02880
 LIS02890
 LIS02900
 LIS02910
 LIS02920
 LIS02930
 LIS02940
 LIS02950
 LIS02960
 LIS02970
 LIS02980
 LIS02990
 LIS03000
 LIS03010
 LIS03020
 LIS03030
 LIS03040

FILE LISTSM

```

DUMA = DOTLAB(J,4,2)
IF (DUMA.EQ.NAMECT) N21 = N21 + 1
IF (DUMA.EQ.SYMDES) GO TO 554
IF (DUMA.EQ.SYMTHR) GO TO 554
IF (DUMA.EQ.SYMOUT) GO TO 554
IF (DUMA.NE.NAMECT) N22 = N22 + 1
554 CONTINUE
555 CONTINUE
580 CONTINUE

C
C*** COMPUTE PROPORTION OF PREFERRED CLASS
C
C23456789
P11 = FLOAT(N11)/FLOAT(N11 + N21 + NB1)
P12 = FLOAT(N12)/FLOAT(N12 + N22 + NB2)
PB1 = FLOAT(NB1)/FLOAT(N11 + N21 + NB1)
PB2 = FLOAT(NB2)/FLOAT(N12 + N22 + NB2)
P = PMACH(1)*(P11 + ALP(1)*PB1)
P = P + PMACH(2)*(P12 + ALP(2)*PB2)

C
WRITE(6,990) ALPMSG(ISIT)
WRITE(6,1000)NAMECT
WRITE(6,1050) PMACH(1),PMACH(2)
WRITE(6,1005)
WRITE(6,1010)N11,N12,N21,N22
WRITE(6,1020)NB1,NB2
WRITE(6,1030)
WRITE(6,1040) P11,P12,PB1,PB2,P,ALP(1),ALP(2)
GO TO 675

C
990 FORMAT(1H1,' TYPE II DOT REPORTS FOR LIST PROCESSING', 'A4,
1 ' VS MACHINE CLASS')
1000 FORMAT(1H0,' PROPORTION SUMMARY FOR CATEGORY 1 = ',1A4)
1005 FORMAT(2X,'CLASS',5X,'1',5X,'2',/,2X,'LABEL')
1010 FORMAT(4X,'1',2X,2(1X,I5),/,4X,'2',2X,2(1X,I5))
1020 FORMAT(4X,'B',2X,2(1X,I5))
1030 FORMAT(1H0,' P11 P12 PB1 PB2 P ALP ')
1040 FORMAT(1H0,7(2X,F6.4))
1050 FORMAT(1H0,' P(1) = ',F7.4,' P(2) = ',F7.4)

C
C CODE ADDED TO PRINT CONFUSION MATRIX N*N
C
582 TOP=NOCAT+1
DO 590 JJ=1,209
LAB=DOTNUM(JJ,1)
IF(LAB.EQ.0)GO TO 590
IF(LAB.EQ.TOP)GO TO 585
LABTOT(LAB)=LABTOT(LAB)+1
CLS=DOTNUM(JJ,2)
IF(CLS.EQ.0)GO TO 590
DOTTOT(CLS)=DOTTOT(CLS)+1
IF(LAB.EQ.TOP)GO TO 590
CONF(LAB,CLS)=CONF(LAB,CLS)+1
590 CONTINUE

C
C WRITE CONFUSION MATRIX
C
2000 WRITE(6,2000)
FORMAT(1H1,2X,'TYPE II DOT REPORTS',/,2X,'CONFUSION MATRIX')
STCAT=1
ENDCAT=NOCAT
IF(ENDCAT.GT.15)ENDCAT=15
TIMES=NOCAT/15
IF(MOD(NOCAT,15).NE.0)TIMES=TIMES+1
DO 595 JJ=1,TIMES
WRITE(6,2010) (CATNAM(KK),KK=STCAT,ENDCAT)
2010 FORMAT(/,3X,'CLASS',5X,1A4,14(2X,1A4))
WRITE(6,2012)ALPMSG(1SIT)
2012 FORMAT(3X,'-----',/,3X,1A4,'LABEL')
DO 610 KK=1,NOCAT
WRITE(6,2020)CATNAM(KK), (CONF(KK,LL),LL=STCAT,ENDCAT)
2020 FORMAT(4X,1A4,15I6)
610 CONTINUE
STCAT=STCAT+15
ENDCAT=NOCAT
IF(ENDCAT.GT.STCAT+14)ENDCAT=STCAT+14

```

LIS03050
 LIS03060
 LIS03070
 LIS03080
 LIS03090
 LIS03100
 LIS03110
 LIS03120
 LIS03130
 LIS03140
 LIS03150
 LIS03160
 LIS03170
 LIS03180
 LIS03190
 LIS03200
 LIS03210
 LIS03220
 LIS03230
 LIS03240
 LIS03250
 LIS03260
 LIS03270
 LIS03280
 LIS03290
 LIS03300
 LIS03310
 LIS03320
 LIS03330
 LIS03340
 LIS03350
 LIS03360
 LIS03370
 LIS03380
 LIS03390
 LIS03400
 LIS03410
 LIS03420
 LIS03430
 LIS03440
 LIS03450
 LIS03460
 LIS03470
 LIS03480
 LIS03490
 LIS03500
 LIS03510
 LIS03520
 LIS03530
 LIS03540
 LIS03550
 LIS03560
 LIS03570
 LIS03580
 LIS03590
 LIS03600
 LIS03610
 LIS03620
 LIS03630
 LIS03640
 LIS03650
 LIS03660
 LIS03670
 LIS03680
 LIS03690
 LIS03700
 LIS03710
 LIS03720
 LIS03730
 LIS03740
 LIS03750
 LIS03760
 LIS03770
 LIS03780
 LIS03790
 LIS03800

FILE LISTSM

ORIGINAL PAGE IS
OF POOR QUALITY

```

595  CONTINUE
C
C  CODE ADDED TO PRINT ALPHA VALUE MATRIX
C
3000  WRITE(6,3000)
      FORMAT(//,2X,'ALPHA VALUE MATRIX')
      STCAT=1
      ENDCAT=NOCAT
      IF(ENDCAT.GT.15)ENDCAT=15
      DO 720 JJ=1,TIMES
        WRITE(6,2010)(CATNAM(KK),KK=STCAT,ENDCAT)
        WRITE(6,2012)ALPMSG(1:ISIT)
        DO 730 KK=1,NOCAT
          DO 740 II=STCAT,ENDCAT
            IF(DOTTOT(II).EQ.0)ALPHA(II-STCAT+1)=0
            IF(DOTTOT(II).EQ.0)GO TO 740
            ALPHA(II-STCAT+1)=CONF(KK,II)/DOTTOT(II)
          CONTINUE
        L=ENDCAT-STCAT+1
        WRITE(6,3020)CATNAM(KK), (ALPHA(II),II=1,L)
      3020  FORMAT(4X,1A,2X,15F6.3)
      730  CONTINUE
          STCAT=STCAT+15
          ENDCAT=NOCAT
          IF(ENDCAT.GT.STCAT+14)ENDCAT=STCAT+14
      720  CONTINUE
C
C  CODE ADDED TO PRINT DOT PERFORMANCE SUMMARY
C
2030  WRITE(6,2030)
      FORMAT(//,2X,'DOT DATA PERFORMANCE SUMMARY')
      WRITE(6,2035)
2035  FORMAT(/,5X,'CATEGORY',9X,'CLASSIFIED',10X,'BIAS CORRECTED ',
1  'PROPORTION',12X,'RANDOM SAMPLE PROPORTION',11X,
2  'VARIANCE')
      WRITE(6,2038)
2038  FORMAT(7X,'NAME',12X,'ESTIMATE',10X,'ESTIMATE',10X,'VARIANCE',
1  10X,'ESTIMATE',10X,'VARIANCE',11X,'RATIO')
C
C  BASE=PIXTOT+TOTALS(NSUB7)
C
C  LOOP TO CALCULATE & PRINT SUMMARY BY CATEGORY
C
DO 650 KK=1,NOCAT
  ALPSUM=0
  DO 625 II=1,NOCAT
    IF(DOTTOT(II).EQ.0)TRM1=0
    IF(DOTTOT(II).EQ.0)GO TO 630
    TRM1=CONF(KK,II)/DOTTOT(II)*TOTCAT(II)
  630  ALPSUM=ALPSUM+TRM1
  625  CONTINUE
  RS=(LABTOT(KK)/SUM)*(PIXTOT/BASE)*100
  VRS=RS*((PIXTOT/BASE)*100-RS)/(SUM-1)
  RC=(ALPSUM/BASE)*100
  VRC=0
  DO 635 LL=1,NOCAT
    TRM1=((TOTCAT(LL)/BASE)*100)**2
    IF(DOTTOT(LL).EQ.0)TRM3=0
    IF(DOTTOT(LL).EQ.0)GO TO 640
    TRM2=CONF(KK,LL)/DOTTOT(LL)
    TRM3=(TRM2*(1-TRM2))/(DOTTOT(LL)-1)
  640  VRC=VRC+TRM1*TRM3
  635  CONTINUE
  PG=VRC/VRS
  CLP=TOTCAT(KK)/PIXTOT*100
  WRITE(6,2040)CATNAM(KK),CLP,BC,VBC,RS,VRS,PG
2040  FORMAT(9X,1A,4X,6(6X,F8.4,4X))
650  CONTINUE
675  WRITE(6,2050)
2050  FORMAT(1H1)
600  CONTINUE
      RETURN
      END

```

LIS03A10
 LIS03A20
 LIS03A30
 LIS03A40
 LIS03A50
 LIS03A60
 LIS03A70
 LIS03A80
 LIS03A90
 LIS03900
 LIS03910
 LIS03920
 LIS03930
 LIS03940
 LIS03950
 LIS03960
 LIS03970
 LIS03980
 LIS03990
 LIS04000
 LIS04010
 LIS04020
 LIS04030
 LIS04040
 LIS04050
 LIS04060
 LIS04070
 LIS04080
 LIS04090
 LIS04100
 LIS04110
 LIS04120
 LIS04130
 LIS04140
 LIS04150
 LIS04160
 LIS04170
 LIS04180
 LIS04190
 LIS04200
 LIS04210
 LIS04220
 LIS04230
 LIS04240
 LIS04250
 LIS04260
 LIS04270
 LIS04280
 LIS04290
 LIS04300
 LIS04310
 LIS04320
 LIS04330
 LIS04340
 LIS04350
 LIS04360
 LIS04370
 LIS04380
 LIS04390
 LIS04400
 LIS04410
 LIS04420
 LIS04430
 LIS04440
 LIS04450
 LIS04460
 LIS04470
 LIS04480
 LIS04490
 LIS04500
 LIS04510
 LIS04520
 LIS04530

FILE: MAPHD

* SUBROUTINE MAPHD(NOCAT,CLSSYM,CATNAM,KATNO,CLSMTX,SUBNO,
SUBDES,CLSV2,NOCLS2,NOSUB2,THRSKY,THRES)

THIS ROUTINE PRINTS THE HEADER INFORMATION FOR THE CLASSIFICATION
MAP IN DISPLAY

NOCAT -- NO. OF CATEGORIES
CLSSYM -- SYMBOLS FOR CATEGORIES OR SURCLASSES
CATNAM -- CATEGORY NAMES
KATNO -- CATEGORY EACH CLASS WAS ASSIGNED TO
CLSMTX -- CLASS NAMES
SUBNO -- NO. OF SURCLASSES IN EACH CLASS
SUBDES -- SURCLASS NAMES
CLSV2 -- CLASS EACH SUBCLASS WAS ASSIGNED TO (IN COMMON
BLOCK INFORM)

IMPLICIT INTEGER (A-Z)

DIMENSION CLSV2(1)
REAL THRES(1)

LOGICAL ISWTH
DIMENSION CLSSYM(1),CATNAM(1),KATNO(1),CLSMTX(1),SUBNO(1),
SUBDES(1)

PRINTS CATEGORY CLASSIFIER INFORMATION

IF (NOCAT .LE. 0) GO TO 82

WRITE(6,200)

200 FORMAT(// T42,'MAP OF CATEGORY CLASSIFIER CLASSIFICATION RESULTS',

* /// T32,'CATEGORY',T42,'CLASS',T93,'SURCLASS',T31,

* 'NO.',T37,'NAME',

* T44,'NO.',T94,'NAME',T101,'SYMBOL',

IF (THRSKY .NE. 0) WRITE(6,205)

205 FORMAT(1H+,T108,'THRES.')

DO 6P I=1,NOCAT

WRITE(6,210)I,CATNAM(I)

210 FORMAT(/T31,I2,T37,A4)

ISWTH = .TRUE.

DO 63 J=1,NOCLS2

IF (KATNO(J) .EQ. I) GO TO 64

GO TO 63

64 IF (ISWTH) GO TO 65

WRITE(6,220)J,CLSMTX(J)

220 FORMAT(/ T44,I2,T66,A4)

ISWTH = .TRUE.

GO TO 66

65 WRITE(6,230)J,CLSMTX(J)

230 FORMAT(1H+,T40,I2,T66,A4)

66 DO 67 K=1,NOSUB2

IF (CLSV2(K) .EQ. J) GO TO 70

GO TO 67

70 NSURCL = SUBNO(J)

KK = 0

DO 75 L=1,NSURCL

KK = K + L - 1

IF (ISWTH) GO TO 72

WRITE(6,250)KK,SUBDES(KK),CLSSYM(KK)

250 FORMAT(T48,I2,T94 ,A4,T103,A1)

GO TO 74

72 WRITE(6,240)KK,SUBDES(KK),CLSSYM(KK)

240 FORMAT(1H+,T44,I2,T94,A4,T103,A1)

ISWTH = .FALSE.

IF (THRSKY .NE. 0) WRITE(6,245)THRES(KK)

245 FORMAT(1H+,T108,F6.3)

75 CONTINUE

GO TO 63

67 CONTINUE

63 CONTINUE

6A CONTINUE

RETURN

PRINTS STANDARD CLASSIFIER INFORMATION

MAP00010
MAP00020
MAP00030
MAP00040
MAP00050
MAP00060
MAP00070
MAP00080
MAP00090
MAP00100
MAP00110
MAP00120
MAP00130
MAP00140
MAP00150
MAP00160
MAP00170
MAP00180
MAP00190
MAP00200
MAP00210
MAP00220
MAP00230
MAP00240
MAP00250
MAP00260
MAP00270
MAP00280
MAP00290
MAP00300
MAP00310
MAP00320
MAP00330
MAP00340
MAP00350
MAP00360
MAP00370
MAP00380
MAP00390
MAP00400
MAP00410
MAP00420
MAP00430
MAP00440
MAP00450
MAP00460
MAP00470
MAP00480
MAP00490
MAP00500
MAP00510
MAP00520
MAP00530
MAP00540
MAP00550
MAP00560
MAP00570
MAP00580
MAP00590
MAP00600
MAP00610
MAP00620
MAP00630
MAP00640
MAP00650
MAP00660
MAP00670
MAP00680
MAP00690
MAP00700
MAP00710
MAP00720
MAP00730
MAP00740
MAP00750
MAP00760
MAP00770
MAP00780
MAP00790

FILE: MAPMD

| | | |
|-----|---|----------|
| A2 | CONTINUE | MAP00800 |
| C | | MAP00810 |
| | WRITE(A,260) | MAP00820 |
| 260 | FORMAT(// T42, 'MAP OF STANDARD CLASSIFIER CLASSIFICATION RESULT | MAP00830 |
| | *S' ///T45, 'CLASS', T77, 'SURCLASS' / T42, 'NO.', T50, 'NAME', T72, 'NO.', | MAP00840 |
| | *T78, 'NAME', T85, 'SYMBOL') | MAP00850 |
| | IF (THRSKY.NE.0) WRITE(6,265) | MAP00860 |
| 265 | FORMAT(1H+, T93, 'THRES.') | MAP00870 |
| C | | MAP00880 |
| | CLSNUM = 1 | MAP00890 |
| | ISWTH = .TRUE. | MAP00900 |
| | DO A9 I=1, NOSI1H2 | MAP00910 |
| | IF (CLSNUM.F3. CLSVC2(I)) GO TO 85 | MAP00920 |
| | CLSNUM = CLSNUM + 1 | MAP00930 |
| | GO TO 87 | MAP00940 |
| 85 | IF (ISWTH) GO TO 87 | MAP00950 |
| | WRITE(6,270) I, SURDES(I), CLSSYM(I) | MAP00960 |
| 270 | FORMAT(T72, I2, T78, A4, T87, A1) | MAP00970 |
| | GO TO 86 | MAP00980 |
| 87 | WRITE(6,280) CLSNUM, CLSMTX(CLSNUM), I, SURDES(I), CLSSYM(I) | MAP00990 |
| 280 | FORMAT(/ T42, I2, T50, A4, T72, I2, T78, A4, T87, A1) | MAP01000 |
| | ISWTH = .FALSE. | MAP01010 |
| 88 | IF (THRSKY.NE.0) WRITE(6,285) THRES(I) | MAP01020 |
| 285 | FORMAT(1H+, T93, F6.3) | MAP01030 |
| 89 | CONTINUE | MAP01040 |
| | RETURN | MAP01050 |
| | END | MAP01060 |

FILE: PCT

```

SUBROUTINE PCT(LINUM,IR,FIELD,VERTEX,FLOSAV,PCTAB,NOFLD,
               SAMSTR,SAMEND,SAMINC)
  IMPLICIT INTEGER(A-Z)

  THIS SUBROUTINE BUILDS THE PERFORMANCE TABLE FOR DISPLAY. OR THE
  HISTOGRAM OF THE QUADRATIC FORM FOR EMPIRICAL THRESHOLDS.
  ARGUMENTS:
    LINUM - LINE NO. BEING TESTED.
    PTS   - NO. OF POINTS IN IR ARRAY
    FIELD - RECTANGULAR COORDINATES OF FIELDS (TRAINING OR TEST)
             (5,NOFLD)
             1-LINE START
             2-LINE END
             3-SAMPLE START
             4-SAMPLE END
             5-POINTER TO VERTEX ARRAY FOR VERTICES OF THIS FIELD
    VERTEX - ARRAY CONTAINING VERTICES FOR ALL FIELDS
    FLOSAV - FIELD INFORMATION
             (4,NOFLD)
             1-FIELD NAME
             2-CLASS NO.
             3-SUBCLASS NO.
             4-NO. OF VERTICES
    PCTAB - PERFORMANCE TABLE
             (NOFLD,NOS(1F3))
    NOFLD - NO. OF FIELDS TO TEST
    IR    - ARRAY CONTAINING THE SUBCLASS NUMBERS FOR POINTS
             ON THIS LINE.
    SAMSTR - BEGINNING SAMPLE NO. OF CLASSIFIED FIELD.
    SAMEND - LAST SAMPLE NO. OF CLASSIFIED FIELD.
    SAMINC - SAMPLE INCREMENT USED IN CLASSIFY

  DIMENSION FIELD(5,NOFLD),VERTEX(1),FLOSAV(4,NOFLD),
             PCTAB(NOFLD,1),IR(1)
  DIMENSION VEC(50),FL(22)
  OPT=1
  GO TO 5
  ENTRY PCTT(LINUM,IR,VR,FIELD,VERTEX,FLOSAV,DSFUNC,NOFLD,
             SAMSTR,SAMEND,SAMINC,CON,RANGE)
  PEAL DSFUNC(RANGE,50),VR(1),CON(1)
  OPT=2
5  CONTINUE

  FIND NUMBER OF FIELDS THAT THIS LINE INTERSECTS
  -----PCTT0620
  -----PCTT0630
  II=0
  DO 10 I=1,NOFLD
    IF (LINUM.LT.FIELD(1,I)) GO TO 10
    IF (LINUM.GT.FIELD(2,I)) GO TO 10
    IF (FIELD(3,I).GT.SAMEND) GO TO 10
    IF (FIELD(4,I).LT.SAMSTR) GO TO 10
    II = II+1
  VEC(II) = I
10  CONTINUE

  NOW CHECK THE FIELDS OF INTEREST ( GIVEN BY 'VEC' )
  -----PCTT0740
  -----PCTT0750
  -----PCTT0760
  -----PCTT0770
  IF (II.EQ. 0) GO TO 35
  DO 30 J=1,II
    JJ = VEC(J)
    NV=FLOSAV(4,JJ)
    IPT=FIELD(5,JJ)
    FIND INTERCEPTS FOR THIS FIELD
    CALL FDLINT(VERTEX(IPT),NV,FL,LINUM,SAMPS,NI)
    DO 30 I=1,1,2
    JB = (FL(I)-SAMSTR)/SAMINC + 1
    JE = (FL(I+1)-SAMSTR)/SAMINC + 1
    IF(MOD(SAMSTR,SAMINC).NE.MOD(FL(I),SAMINC))JB=JB+1
    IF(JB.GT.JE)GO TO 30
    DO 20 K=JB,JE
    K2 = IR(K)
    IF(K2.EQ.0)GO TO 20
    GO TO (15,18),OPT
15  PCTAB(JJ,K2) = PCTAB(JJ,K2) + 1
    GO TO 20
18  CONTINUE
  -----PCTT0790
  -----PCTT0800
  -----PCTT0870

```

FILE: PCT

```
      DIVSN = FLDSAV(3,JJ)
C*      WAS THIS PIXEL CLASSIFIED CORRECTLY
C*
C*      IF(DIVSN.NE.K?)GO TO 20
21  L=10*(-.5*CON(K2)-VR(K)) + 0.5
      IF (L.LE.0) L = 1
      IF (L.GT.RANGE) L = RANGE
      DSFUNC(L,DIVSN) = DSFUNC(L,DIVSN) + 1.0
20  CONTINUE
30  CONTINUE
35  CONTINUE
      RETURN
C
C
C      GO HOME
-----
      END
```

PCTT0900

PCTT0910

PCTT0920

PCTT0930

PCTT0940

PCTT0950

PCTT0970

ORIGINAL PAGE IS
OF POOR QUALITY

FILE: PRTPCT

```
SUBROUTINE PRTPCT (FLDSAV, PCTAB, NOFLD)
PRTPCT PRINTS THE FOLLOWING CLASSIFICATION PERFORMANCE TABLES
1. FIELD BY SUBCLASS - IF PCFDKY=1
2. FIELD BY CLASS - IF PCFDKY=1
3. FIELD BY CATEGROY - IF PCFDKY=1 AND CATFLG=1
4. CLASS BY SURCLASS
5. CLASS BY CLASS
6. CLASS BY CATEGORY - IF CATFLG=1

IMPLICIT INTEGER (A-Z)
REAL PCTT
REAL PCT
DIMENSION FLDSAV (4, NOFLD), PCTAB (NOFLD, NOSUR3), TOTSAM (200), BUF (60)
INCLUDE CMRK6.LIST
INCLUDE CMRK10.LIST
COMMON/GLOBAL/HEAD (63), MAPTAP, DATAPE, SAVTAP, BMFILE, BMKEY,
* HISFIL, HISKEY, TRFORM, ERIPTP, ERPKEY, MAPUNT, NOFILE,
* DRUMAD, DRMWDS, PAGESIZ, DATFIL, STAFIL, ASAV, ASAVFL
* NHSTUN, NHSTFI, SCTRUN, MAPFIL
* DOTUNT, DOTFIL, NCHPAS, TRANSFL, BMTRFL, HISTFL, PCHUNT,
* CRDUNT, PRTUNT, RANDIO
COMMON/DISPL/CATFLG, CATNAM (61), CLSNAM (61), SURNAM (61), SURNO (60),
* SURCAT (60), CLSSUR (60), NOMAP, TOTVT3, NOSUR3,
* PCFDKY, ISTKEY, TRNKEY, THRSKY, STATKY, EMPTRS, THRSVA,
* PLTKY, BMFLG, RMCOMB, BMFEAT, COATE (2),
* FLDSV2, FIELD2, VERTX2, FLDSV3, FIELD3, VERTX3, PCTID3,
* THRES (60), SYMMTX (66), HIGH (60), CON (60)
* , FLDKEY, NOFLD2, NOFLD3, NOFET2, FETVC2 (30)
* , NOSUR2, NOTKFD, TOTVT2, NOCLS2
* , KATNO (60), NOCAT, FILTER, MAPFMT
* , DESKEY, DESUNI, DESOTH, CROP, ACROP, AOTHER, ATOTAL
* , SITE (6), ANALYS (5), CAM (15), CRPKEY, KEPPTS (60)
* , DOTKEY, DOTERR
CSEND
DIMENSION TABLE (60, 61)
DATA THR/THRE/
SURNAM (NOSUR3) = THR
CLSNAM (NOCLS2+1) = THR
CATNAM (NOCAT+1) = THR
IF (PCFDKY.NE.1) GO TO #1
CLASSIFICATION SUMMARIES BY FIELD
SURCLASS PERFORMANCE
IR=1
IE=14
5 IF (IE.GT. NOSUR3) IE=NOSUR3
WRITE (6, HEAD)
IF (ISTKEY.EQ.1) WRITE (6, 100)
IF (ISTKEY.NE.1) WRITE (6, 200)
IF (FLDKEY.EQ.1) WRITE (6, 350)
IF (FLDKEY.NE.1) WRITE (6, 351)
WRITE (6, 300) (SURNAM (I), I=IR, IE)
WRITE (6, 351)
PCTT=0.0
DO 20 J=1, NOFLD
IS=FLDSAV (3, J)
IC=FLDSAV (2, J)
IF (IS.NE.0) NAME=SURNAM (IS)
IF (IS.EQ.0) NAME=CLSNAM (IC)
TOTSAM (J)=0
DO 10 K=1, NOSUR3
10 TOTSAM (J) = TOTSAM (J) + PCTAB (J, K)
IF (FLDKEY.NE.1) GO TO 15
PCT=FLOAT (PCTAB (J, IS))/FLOAT (TOTSAM (J))*100.
PCTT=PCTT+PCT
WRITE (6, 400) FLDSAV (1, J), NAME, TOTSAM (J), PCT, (PCTAB (J, K), K=IR, IF)
GO TO 20
15 WRITE (6, 500) FLDSAV (1, J), NAME, TOTSAM (J), (PCTAB (J, K), K=IR, IE)
20 CONTINUE
IF (IE.EQ. NOSUR3) GO TO 30
IR=IR+1
IF=IF+14
PCTT=0.0
GO TO 5
30 CONTINUE
PCTT=PCTT/NOFLD
```

PRT00010
PRT00020
PRT00030
PRT00040
PRT00050
PRT00060
PRT00070
PRT00080
PRT00090
PRT00100
PRT00110
PRT00120
PRT00130
PRT00140
PRT00150
PRT00160
PRT00170
PRT00180
PRT00190
PRT00200
PRT00210
PRT00220
PRT00230
PRT00240
PRT00250
PRT00260
PRT00270
PRT00280
PRT00290
PRT00300
PRT00310
PRT00320
PRT00330
PRT00340
PRT00350
PRT00360
PRT00370
PRT00380
PRT00390
PRT00400
PRT00410
PRT00420
PRT00430
PRT00440
PRT00450
PRT00460
PRT00470
PRT00480
PRT00490
PRT00500
PRT00510
PRT00520
PRT00530
PRT00540
PRT00550
PRT00560
PRT00570
PRT00580
PRT00590
PRT00600
PRT00610
PRT00620
PRT00630
PRT00640
PRT00650
PRT00660
PRT00670
PRT00680
PRT00690
PRT00700
PRT00710
PRT00720
PRT00730
PRT00740
PRT00750
PRT00760
PRT00770
PRT00780
PRT00790

ORIGINAL PAGE IS
OF POOR QUALITY

FILE: PRTPT

```

      IF (FLOKEY.EQ.1) WRITE(6,860) PCTT
      PCTT=0.0
      IR=1
      IE=14
C*   NOW FIELD BY CLASS
      IN = NOSUR2+1
      32 IF (IE.GT.NOCLS2+1) IE=NOCLS2+1
         WRITE(6,HEAD)
         IF (TSTKEY.NF.1) WRITE(6,600)
         IF (TSTKEY.EQ.1) WRITE(6,650)
         WRITE(6,351)
         WRITE(6,300) (CLSNAM(I),I=IR,IE)
         WRITE(6,351)
C*   DO 50 J=1,NOFLD
      CONDENSE A LINE
      DO 33 K=1,IN
      33 RUF(K)=0
         DO 35 K=1,NOSUR2
            IK=CLSSUR(K)
      35 RUF(IK) = RUF(IK) + PCTAB(J,K)
            IC=FLDSAV(2,J)
            RUF(NOCLS2+1) = PCTAB(J,NOSUB3)
            PCT = FLOAT(RUF(IC))/FLOAT(TOTSAM(J)) * 100.
            PCTT=PCTT+PCT
            WRITE(6,400) FLDSAV(1,J),CLSNAM(IC),TOTSAM(J),PCT,
              (RUF(K),K=IR,IE)
      50 CONTINUE
      IF (IF.FQ.NOCLS2+1) GO TO 60
      IR=IF+1
      IE=IF+14
      PCTT=0.0
      GO TO 32
      60 CONTINUE
      PCTT=PCTT/NOFLD
      WRITE(6,860) PCTT
      PCTT=0.0
C*   C*   C*
C*   NOW FIELD BY CATEGORY
C*   IF (CATFLG.EQ.0) GO TO A1
      IR=1
      IE=IE+14
      62 IF (IF.GT.NOCAT+1) IE=NOCAT + 1
         WRITE(6,HEAD)
         IF (TSTKEY.NF.1) WRITE(6,700)
         IF (TSTKEY.FQ.1) WRITE(6,750)
         WRITE(6,352) (CATNAM(I),I=IR,IE)
         WRITE(6,351)
         DO 70 J=1,NOFLD
            DO 63 K=1,IN
      63 RUF(K)=0
         DO 65 K=1,NOSUR2
            IK = SURCAT(K)
      65 RUF(IK) = RUF(IK) + PCTAB(J,K)
            IC=FLDSAV(2,J)
            ICAT=KATNO(IC)
            PCT = FLOAT(RUF(ICAT))/FLOAT(TOTSAM(J)) * 100.
            PCTT=PCTT+PCT
            RUF(NOCAT+1) = PCTAB(J,NOSUB3)
            WRITE(6,400) FLDSAV(1,J),CATNAM(ICAT),TOTSAM(J),PCT,
              (RUF(K),K=IR,IF)
      70 CONTINUE
      IF (IE.FQ.NOCAT+1) GO TO A0
      IR=IF+1
      IE=IF+14
      PCTT=0.0
      GO TO 62
      A0 CONTINUE
      PCTT=PCTT/NOFLD
      WRITE(6,860) PCTT
C*   C*   C*
C*   NOW COMPRESS PCTAB TO CLASS BY SUBCLASS
      ZERO TABLE
      A1 CONTINUE
      DO 85 I=1,NOCLS2
         TOTSAM(I)=0
      DO 85 J=1,NOSUB3
      85 TABLE(I,J)=0

```

PRT00800
 PRT00810
 PRT00820
 PRT00830
 PRT00840
 PRT00850
 PRT00860
 PRT00870
 PRT00880
 PRT00890
 PRT00900
 PRT00910
 PRT00920
 PRT00930
 PRT00940
 PRT00950
 PRT00960
 PRT00970
 PRT00980
 PRT00990
 PRT01000
 PRT01010
 PRT01020
 PRT01030
 PRT01040
 PRT01050
 PRT01060
 PRT01070
 PRT01080
 PRT01090
 PRT01100
 PRT01110
 PRT01120
 PRT01130
 PRT01140
 PRT01150
 PRT01160
 PRT01170
 PRT01180
 PRT01190
 PRT01200
 PRT01210
 PRT01220
 PRT01230
 PRT01240
 PRT01250
 PRT01260
 PRT01270
 PRT01280
 PRT01290
 PRT01300
 PRT01310
 PRT01320
 PRT01330
 PRT01340
 PRT01350
 PRT01360
 PRT01370
 PRT01380
 PRT01390
 PRT01400
 PRT01410
 PRT01420
 PRT01430
 PRT01440
 PRT01450
 PRT01460
 PRT01470
 PRT01480
 PRT01490
 PRT01500
 PRT01510
 PRT01520
 PRT01530
 PRT01540
 PRT01550
 PRT01560
 PRT01570
 PRT01580

FILE: PRTPCT

```

DO 90 I=1,NOFLO
IC=FLOSAV(2,1)
DO 90 J=1,NOSUB3
TABLE(IC,J)=TABLE(IC,J)+PCTAB(I,J)
TOTSAM(IC)=TOTSAM(IC)+PCTAB(I,J)
C* 90 CONTINUE
CLASS BY SURCLASS
IR=1
IE=14
91 IF (IF.GT.NOSUB3) IE=NOSUB3
WRITE(6,HEAD)
IF (TSTKEY.NE.1) WRITE(6,800)
IF (TSTKEY.EQ.1) WRITE(6,85)
WRITE(6,A10) (SURNAM(I),I=IR,IE)
WRITE(6,351)
DO 92 I=1,NOCLS2
IF (TOTSAM(I).EQ.0) GO TO 92
WRITE(6,820) CLSNAM(I),TOTSAM(I),(TABLE(I,J),J=IR,IE)
92 CONTINUE
IR=IR+14
IE=IE+14
IF (IE-14 .LT. NOSUB3) GO TO 91
C*
C* CLASS BY CLASS
DO 93 I=1,NOCLS2
DO 97 J=1,NOCLS2
97 RUF(J)=0
DO 94 J=1,NOSUR2
IC=CLASSUR(J)
RUF(IC)=RUF(IC) + TABLE(I,J)
DO 95 J=1,NOCLS2
TABLE(I,J)=RUF(J)
TABLE(I,NOCLS2+1)=TABLE(I,NOSUB3)
93 CONTINUE
PCTT=0
IR=1
IE=14
96 IF (IF.GT.NOCLS2+1) IE=NOCLS2+1
WRITE(6,HEAD)
IF (TSTKEY.EQ.1) WRITE(6,830)
IF (TSTKEY.NE.1) WRITE(6,835)
WRITE(6,A10) (CLSNAM(I),I=IR,IE)
WRITE(6,351)
NC=0
DO 107 I=1,NOCLS2
IF (TOTSAM(I).EQ.0) GO TO 107
NC=NC+1
PCT=(FLOAT(TABLE(I,I))/FLOAT(TOTSAM(I)))*100.
PCTT=PCTT + PCT
WRITE(6,850) CLSNAM(I),TOTSAM(I),PCT,(TABLE(I,J),J=IR,IE)
107 CONTINUE
IF (IF.EQ.NOCLS2+1) GO TO 108
IR=IR+14
IE=IE+14
PCTT=0.0
GO TO 96
108 CONTINUE
PCTT=PCTT/NC
WRITE(6,860) PCTT
C*
C* NOW CLASS BY CATEGORY
IF (CATFLG.EQ.0) RETURN
DO 116 J=1,NOCLS2
DO 112 I=1,NOCAT
112 RUF(I)=0
DO 114 K=1,NOCLS2
IC=CATNO(K)
RUF(IC)=RUF(IC) + TABLE(J,K)
DO 115 K=1,NOCAT
TABLE(J,K)=RUF(K)
TABLE(J,NOCAT+1)=TABLE(J,NOCLS2+1)
116 CONTINUE
PCTT=0.0
IR=1
IE=14
117 IF (IF.GT.NOCAT+1) IE=NOCAT+1
WRITE(6,HEAD)

```

PRT01590
 PRT01600
 PRT01610
 PRT01620
 PRT01630
 PRT01640
 PRT01650
 PRT01660
 PRT01670
 PRT01680
 PRT01690
 PRT01700
 PRT01710
 PRT01720
 PRT01730
 PRT01740
 PRT01750
 PRT01760
 PRT01770
 PRT01780
 PRT01790
 PRT01800
 PRT01810
 PRT01820
 PRT01830
 PRT01840
 PRT01850
 PRT01860
 PRT01870
 PRT01880
 PRT01890
 PRT01900
 PRT01910
 PRT01920
 PRT01930
 PRT01940
 PRT01950
 PRT01960
 PRT01970
 PRT01980
 PRT01990
 PRT02000
 PRT02010
 PRT02020
 PRT02030
 PRT02040
 PRT02050
 PRT02060
 PRT02070
 PRT02080
 PRT02090
 PRT02100
 PRT02110
 PRT02120
 PRT02130
 PRT02140
 PRT02150
 PRT02160
 PRT02170
 PRT02180
 PRT02190
 PRT02200
 PRT02210
 PRT02220
 PRT02230
 PRT02240
 PRT02250
 PRT02260
 PRT02270
 PRT02280
 PRT02290
 PRT02300
 PRT02310
 PRT02320
 PRT02330
 PRT02340
 PRT02350
 PRT02360
 PRT02370

ORIGINAL PAGE IS
OF POOR QUALITY

FILE: PRTPCT

```
IF (TSTKEY.EQ.1) WRITE (6,870)
IF (TSTKEY.NE.1) WRITE (6,875)
WRITE (6,810) (CATNAM(I), I=1B, 1E)
WRITE (6,351)
NC=0
DO 11A I=1, NOCLS2
IF (TOTSAM(I).EQ.0) GO TO 118
NC=NC+1
IC=KATNO(I)
PCT=(FLOAT(TABLE(I,IC))/FLOAT(TOTSAM(I)))*100.
PCTT=PCTT + PCT
WRITE (6,850) CLSNAM(I), TOTSAM(I), PCT, (TABLE(I,J), J=1B, 1E)
118 CONTINUE
IF (IF.GE.NOCAT+1) GO TO 119
IF=IF+14
IE=IF+14
PCTT=0.0
GO TO 117
119 CONTINUE
PCTT=PCTT/NC
WRITE (6,860) PCTT
FORMAT (10X, 'CLASSIFICATION SUMMARY - TEST FIELDS BY SURCLASS')
200 FORMAT (10X, 'CLASSIFICATION SUMMARY - TRAINING FIELDS BY SUBCLASS')
300 FORMAT (1H, 'FIELD', T17, 'TOTAL', T24, 'PCT', /
* 1X, 'NAME', T10, 'CLASS', T17, 'PTS', T24, 'CORCT', .2X, 14 (1X, A6) /)
351 FORMAT (/)
352 FORMAT (1X, 'FIELD', T17, 'TOTAL', T24, 'PCT', /
* 1X, 'NAME', T10, 'CAT', T17, 'PTS', T24, 'CORCT', .2X, 14 (1X, A6) /)
350 FORMAT (1H, 'SUR')
400 FORMAT (3X, A4, 3X, A4, 15, 2X, F7.2, 2X, 14 (15, 2X))
500 FORMAT (3X, A4, 3X, A4, 15, 12X, 14 (15, 2X))
600 FORMAT (10X, 'CLASSIFICATION SUMMARY - TRAINING FIELDS BY CLASS')
700 FORMAT (10X, 'CLASSIFICATION SUMMARY - TEST FIELDS BY CLASS')
800 FORMAT (10X, 'CLASSIFICATION SUMMARY - TRAINING FIELDS BY CATEGORY')
900 FORMAT (10X, 'CLASSIFICATION SUMMARY - TEST FIELDS BY CATEGORY')
A00 FORMAT (10X, 'CLASSIFICATION SUMMARY - TRAINING CLASS BY SUBCLASS')
A10 FORMAT (10X, 'CLASSIFICATION SUMMARY - TEST CLASS BY SUBCLASS')
* 5X, 14 (3X, A4) /)
A20 FORMAT (3X, A4, 3X, 15, 15X, 14 (15, 2X))
A30 FORMAT (10X, 'CLASSIFICATION SUMMARY - TEST CLASS BY CLASS')
A40 FORMAT (10X, 'CLASSIFICATION SUMMARY - TRAINING CLASS BY CLASS')
A50 FORMAT (3X, A4, 3X, 15, 3X, F7.2, 5X, 14 (15, 2X))
A60 FORMAT (10X, 'CLASSIFICATION SUMMARY - TEST CLASS BY SURCLASS')
A70 FORMAT (10X, 'CLASSIFICATION SUMMARY - TRAINING CLASS BY SURCLASS')
A80 FORMAT (10X, 'CLASSIFICATION SUMMARY - TEST CLASS BY CATEGORY')
A90 FORMAT (10X, 'CLASSIFICATION SUMMARY - TRAINING CLASS BY CATEGORY')
875 FORMAT (10X, 'CLASSIFICATION SUMMARY - TEST CLASS BY CATEGORY')
876 FORMAT (10X, 'CLASSIFICATION SUMMARY - TRAINING CLASS BY CATEGORY')
RETURN
END
```

PRT023A0
PRT02390
PRT02400
PRT02410
PRT02420
PRT02430
PRT02440
PRT02450
PRT02460
PRT02470
PRT02480
PRT02490
PRT02500
PRT02510
PRT02520
PRT02530
PRT02540
PRT02550
PRT02560
PRT02570
PRT02580
PRT02590
PRT02600
PRT02610
PRT02620
PRT02630
PRT02640
PRT02650
PRT02660
PRT02670
PRT02680
PRT02690
PRT02700
PRT02710
PRT02720
PRT02730
PRT02740
PRT02750
PRT02760
PRT02770
PRT02780
PRT02790
PRT02800
PRT02810
PRT02820
PRT02830
PRT02840
PRT02850

FILE: PRISUM

```

SUBROUTINE PRISUM(TOTALS,TTOL,FLDESC)
  INCLUDE CMRK10.LIST
  INCLUDE COMRK6.LIST
  COMMON/GLOBAL/HEAD(63),MAPTAP,DATAPE,SAVTAP,BMFILE,BMKEY,
    HTSFIL,HISKEY,TRFORM,EMPTP,ERPKEY,MAPUNT,NOFILE,
    DRUMAD,DRMWDS,PAGSIZ,DATFIL,STAFIL,ASAV,ASAVFL,
    NMSTLN,NMSTFI,SCTRUN,MAPFIL,
    DOTUNT,DOTFIL,NCHPAS,TRANSFL,BMTRFL,HISTFL,PCMUNT,
    CRUMNT,PRUNT,RAND10
  COMMON/DISPL/CATFLG,CATNAM(61),CLSNAM(61),SUBNAM(61),SURN(60),
    SURCAT(60),CLSSUB(60),NOMAP,TOTVT3,NOSUB3,
    PCFKY,TSTKEY,TANKEY,THRSKY,STATKY,EMPTRS,THRSVA,
    PLTKEY,BMFLG,BMCOMB,BMFEAT,COATE(2),
    FLOSV2,FIELD2,VENTX2,FLOSV3,FIELD3,VENTX3,PCTID3,
    THRS(60),SYMMTX(66),HIGH(60),CON(60),
    FLDKEY,NOFLD2,NOFLD3,NOFET2,FETVC2(30),
    NOSUR2,NOTHED,TOTVT2,NOCLS2,
    KATNO(60),NOCAT,FILTER,MAPFMT,
    DESKEY,DESUNI,DESOTH,CROP,ACROP,AOTHER,ATOTAL,
    SITF(4),ANALYS(5),CAM(15),CRPKEY,KEPPTS(60),
    NOTKEY,DOTERR
  CSEND
  DIMENSION TOTALS(66),TTOL(66)
  INTEGER G, CRPTYP,F,M,Z,Y
  INTEGER TTOL,DESUNI,DESOTH
  INTEGER TIME,CLSSUB,CATFLG,SUBCAT
  INTEGER CRPKEY,CROP,CATNAM,CLSNAM,SUBNAM
  C*
  C* IF INTENSIVE TEST SITE SUMMARY REPORT IS TO BE PRINTED, FIND A
  C* MATCH ON 'CROP' NAME.
  C*
  CRPTYP=0
  IF(CRPKEY.NE.1)GO TO 10
  DO 2 I=1,NOCAT
  IF(CROP.EQ.CATNAM(I))GO TO 6
  2 CONTINUE
  DO 3 I=1,NOCLS2
  IF(CROP.EQ.CLSNAM(I))GO TO 7
  3 CONTINUE
  DO 4 I=1,NOSUR2
  IF(CROP.EQ.SURNAM(I))GO TO 8
  4 CONTINUE
  WRITE(A,490)CROP
  490 FORMAT(' THE CROP NAME ',A4,' DOES NOT MATCH A CATEGORY,CLASS OR
  SUBCLASS NAME.'/) THE INTENSIVE TEST SITE SUMMARY REPORT CANNOT BE
  PRINTED)
  CRPKEY=0
  GO TO 10
  C*
  C* CROP IS A CATEGORY      CRPTYP=1
  C* CROP IS A CLASS        CRPTYP=2
  C* CROP IS A SUBCLASS     CRPTYP=3
  C*
  6 INDEX=1
  CRPTYP=1
  GO TO 10
  7 INDEX=1
  CRPTYP=2
  GO TO 10
  8 INDEX=1
  CRPTYP=3
  10 CONTINUE
  G=0
  H=0
  I=0
  J=0
  JJ=0
  DO 20 I=1,66
  20 JJ=TOTALS(I) + JJ
  C*
  C* PRINT CLASSIFICATION SUMMARY FOR THIS FIELD
  C*
  CALL SFTMNG(68,4,62)
  WRITE(A,HEAD)
  WRITE(A,260)FLDESC,JJ
  J=JJ - TOTALS(DESUNI)
  KT = TOTALS(DESUNI)
  IF(J.LT. JJ)WRITE(6,265) KT, J
  MT=TOTALS(DESOTH)

```

PR00010
 PR00020
 PR00030
 PR00040
 PR00050
 PR00060
 PR00070
 PR00080
 PR00090
 PR00100
 PR00110
 PR00120
 PR00130
 PR00140
 PR00150
 PR00160
 PR00170
 PR00180
 PR00190
 PR00200
 PR00210
 PR00220
 PR00230
 PR00240
 PR00250
 PR00260
 PR00270
 PR00280
 PR00290
 PR00300
 PR00310
 PR00320
 PR00330
 PR00340
 PR00350
 PR00360
 PR00370
 PR00380
 PR00390
 PR00400
 PR00410
 PR00420
 PR00430
 PR00440
 PR00450
 PR00460
 PR00470
 PR00480
 PR00490
 PR00500
 PR00510
 PR00520
 PR00530
 PR00540
 PR00550
 PR00560
 PR00570
 PR00580
 PR00590
 PR00600
 PR00610
 PR00620
 PR00630
 PR00640
 PR00650
 PR00660
 PR00670
 PR00680
 PR00690
 PR00700
 PR00710
 PR00720
 PR00730
 PR00740
 PR00750
 PR00760
 PR00770
 PR00780
 PR00790

FILE: PRTSUM

ORIGINAL PAGE IS
OF POOR QUALITY

```

IF (MT.GT.0) WRITE (6,266) MT
WRITE (6,270)
WRITE (6,275)
WRITE (6,288)
WRITE (6,276)
DO 290 I=1,NOSUR2
  ITTL=TOTALS(I).TTOL(I)
  PCTT=FLOAT(ITTL)/FLOAT(J)*100.
  IT=TOTALS(I)
  PIT=TOTALS(I)/FLOAT(J)*100.
  PT=TTOL(I)
  PC=PT/FLOAT(ITTL)*100.
  PPIT=100.-PC
  PT=PT/FLOAT(J)*100.
  WRITE (6,280) SURNAM(I),ITTL,PCTT,IT,PIT,PPIT,TTOL(I),PT,PC
  IF (CHPTYP.NE.3) GO TO 290
  IF (INDEX.EQ.1) GO TO 80
  H=H+ITTL
  J=J+IT
  GO TO 290
290 G=G+ITTL
  I=I+1
  CONTINUE
  KTTL=TOTALS(NOSUR3)
  JTTL = TOTALS(NOSUR3)-TTOL(NOSUR3)
  PPCTT=TOTALS(NOSUR3)/FLOAT(J)*100.
  WRITE (6,285) JTTL,TTOL(NOSUR3),KTTL,PPCTT
  TIME=1
  IC=NOCLS2
295 CONTINUE
  WRITE (6,HEAD)
  WRITE (6,260) FLOFSC,JJ
  IF (J.LT.JJ) WRITE (6,265) KT,J
  MT=TOTALS(NESOTH)
  IF (MT.GT.0) WRITE (6,266) MT
  WRITE (6,270)
  IF (TIME.EQ.1) WRITE (6,286)
  IF (TIME.EQ.2) WRITE (6,287)
  WRITE (6,288)
  WRITE (6,276)
  DO 330 IJ=1,IC
    IT=0
    ITT=0
    ITTL=0
    DO 320 I=1,NOSUR2
      IF (TIME.EQ.1) GO TO 315
      IF (SURCAT(I).NE.IJ) GO TO 320
      GO TO 316
    315 IF (CLASSUR(I).NE.IJ) GO TO 320
    316 IT=TOTALS(I)+IT
      ITTL=TTOL(I)+TOTALS(I)*ITTL
      ITT=TTOL(I)+ITT
    320 CONTINUE
      PCTT=FLOAT(ITTL)/FLOAT(J)*100.
      PIT=FLOAT(IT)/FLOAT(J)*100.
      PT=FLOAT(ITT)/FLOAT(J)*100.
      PPIT=FLOAT(IT)/FLOAT(ITTL)*100.
      PC=100.-PPIT
      IF (TIME.EQ.2) GO TO 325
      WRITE (6,280) CLSNAM(IJ),ITTL,PCTT,IT,PIT,PPIT,ITT,PT,PC
      IF (CHPTYP.NE.2) GO TO 326
      IF (INDEX.EQ.1) GO TO 323
      H=H+ITTL
      J=J+IT
      GO TO 326
    323 G=G+ITTL
      I=I+1
      GO TO 326
    325 WRITE (6,280) CATNAM(IJ),ITTL,PCTT,IT,PIT,PPIT,ITT,PT,PC
      IF (CHPTYP.NE.1) GO TO 326
      IF (INDEX.EQ.1) GO TO 324
      H=H+ITTL
      J=J+IT
      GO TO 326
    324 G=G+ITTL
      I=I+1
      CONTINUE
    326 CONTINUE
    330 WRITE (6,285) JTTL,TTOL(NOSUR3),KTTL,PPCTT

```

PRT00800
 PRT00810
 PRT00820
 PRT00830
 PRT00840
 PRT00850
 PRT00860
 PRT00870
 PRT00880
 PRT00890
 PRT00900
 PRT00910
 PRT00920
 PRT00930
 PRT00940
 PRT00950
 PRT00960
 PRT00970
 PRT00980
 PRT00990
 PRT01000
 PRT01010
 PRT01020
 PRT01030
 PRT01040
 PRT01050
 PRT01060
 PRT01070
 PRT01080
 PRT01090
 PRT01100
 PRT01110
 PRT01120
 PRT01130
 PRT01140
 PRT01150
 PRT01160
 PRT01170
 PRT01180
 PRT01190
 PRT01200
 PRT01210
 PRT01220
 PRT01230
 PRT01240
 PRT01250
 PRT01260
 PRT01270
 PRT01280
 PRT01290
 PRT01300
 PRT01310
 PRT01320
 PRT01330
 PRT01340
 PRT01350
 PRT01360
 PRT01370
 PRT01380
 PRT01390
 PRT01400
 PRT01410
 PRT01420
 PRT01430
 PRT01440
 PRT01450
 PRT01460
 PRT01470
 PRT01480
 PRT01490
 PRT01500
 PRT01510
 PRT01520
 PRT01530
 PRT01540
 PRT01550
 PRT01560
 PRT01570
 PRT01580

FILE: PRTSUM

```

IF (TIME.EQ.2) GO TO 340
IF (CATFLG.EQ.0) GO TO 340
IC=NOCAT
TIME=2
GO TO 295
340 CONTINUE
IF (CRPKEY.NE.1) RETURN
WRITE (6,500) CROP,SITE,ANALYS
WRITE (6,505) CAM
D= ACROP/ATOTAL
F= AOTHER/ATOTAL
WRITE (6,510) CROP,ACROP,CROP,D, AOTHER,E, ATOTAL
H=H+TOTALS (DESOTH)
J1=J1+TOTALS (DESOTH)
Z=TOTALS (DESOTH)
Y=J1
F=Y-Z
WRITE (6,520) Y,Z,F,CROP,G,H,CROP,I1,J1
K=G-I1
L=H-J1
M=K+L
WRITE (6,530) CROP,K,L,M
RN=FLOAT (K)/FLOAT (G)
O =FLOAT (L)/FLOAT (H)
P =FLOAT (K)/FLOAT (F)
Q =FLOAT (L)/FLOAT (F)
WRITE (6,540) CROP,RN,O,CROP,P,Q
R=FLOAT (K+L)/FLOAT (F)
S=FLOAT (G)/FLOAT (F)
T=FLOAT (H)/FLOAT (F)
I1=FLOAT (I1)/FLOAT (F)
V=FLOAT (J1)/FLOAT (F)
W=FLOAT (H+K)/FLOAT (F)
WRITE (6,550) R,CROP,S,T,CROP,U,V,W
WRITE (6,610)
WRITE (6,610)
WRITE (6,560)
WRITE (6,575)
WRITE (6,570)
WRITE (6,580)
WRITE (6,570)
WRITE (6,585)
WRITE (6,570)
WRITE (6,590)
WRITE (6,570)
WRITE (6,560)
WRITE (6,595)
WRITE (6,570)
WRITE (6,611)
WRITE (6,570)
T1=S-D
T2=U-D
WRITE (6,600) CROP,G,S,T1,I1,U,P,T2
WRITE (6,570)
WRITE (6,560)
WRITE (6,605)
WRITE (6,570)
WRITE (6,615)
WRITE (6,570)
DATA NAME /'CROP'/
T1=T-E
T2=W-E
KH=H+K
WRITE (6,600) NAME,H,T,T1,KH,W,O,T2
WRITE (6,570)
WRITE (6,570)
WRITE (6,560)
260 FORMAT (// 'CLASSIFICATION SUMMARY FOR FIELD ',A6//
* ' TOTAL NUMBER OF SAMPLED POINTS ',I10)
265 FORMAT (// 'LESS DESIGNATED UNIDENTIFIABLE ',I10/T36,7(' ')/
* T33,I10)
266 FORMAT (// ' NO. OF PIXELS DESIGNATED OTHER',T33,I10//)
500 FORMAT (1H,T44,'INTENSIVE TEST SITE SUMMARY REPORT FOR ',A4 ///
* T20,'NAME OF INTENSIVE TEST SITE ',6A4,T71,'NAME OF ANALYST ',
* 5A4/T42,22(' '),T86,18(' '))
505 FORMAT (T20,'PROCEDURE CONFIGURATION',T48 , 15A4/T48,56(' ')/)
510 FORMAT (T15,'GROUND TRUTH FOR INTENSIVE TEST SITE' /T15,36(' ')/
* T20,'ACHARGE OF ',A4, ' A = ',F6.1,T62,
* 'TRUE PROPORTION IN ',A4,5X,'A/C = D = ',F4.3 /

```

PRT01590
 PRT01600
 PRT01610
 PRT01620
 PRT01630
 PRT01640
 PRT01650
 PRT01660
 PRT01670
 PRT01680
 PRT01690
 PRT01700
 PRT01710
 PRT01720
 PRT01730
 PRT01740
 PRT01750
 PRT01760
 PRT01770
 PRT01780
 PRT01790
 PRT01800
 PRT01810
 PRT01820
 PRT01830
 PRT01840
 PRT01850
 PRT01860
 PRT01870
 PRT01880
 PRT01890
 PRT01900
 PRT01910
 PRT01920
 PRT01930
 PRT01940
 PRT01950
 PRT01960
 PRT01970
 PRT01980
 PRT01990
 PRT02000
 PRT02010
 PRT02020
 PRT02030
 PRT02040
 PRT02050
 PRT02060
 PRT02070
 PRT02080
 PRT02090
 PRT02100
 PRT02110
 PRT02120
 PRT02130
 PRT02140
 PRT02150
 PRT02160
 PRT02170
 PRT02180
 PRT02190
 PRT02200
 PRT02210
 PRT02220
 PRT02230
 PRT02240
 PRT02250
 PRT02260
 PRT02270
 PRT02280
 PRT02290
 PRT02300
 PRT02310
 PRT02320
 PRT02330
 PRT02340
 PRT02350
 PRT02360
 PRT02370

ORIGINAL PAGE IS
OF POOR QUALITY

FILE: PRISUM

```

* T20, 'ACREAGE OF OTHER R = ', F6.1, T62,
* 'TRUE PROPORTION IN OTHER', 4X, 'R/C = E = ', F4.3 /
* T20, 'TOTAL ACREAGE', T39, 'C = ', F6.1 /
520 FORMAT(T15, 'RESULTS OF COMPUTATION FOR INTENSIVE TEST SITE', /
* T15.46(' - ') /
* T20, 'TOTAL NUMBER OF PIXELS IN INTENSIVE TEST SITE', T96, 'Y = ', I6 /
* T20, 'TOTAL NO. OF PIXELS IN EXCLUSION (UNIDENTIFIABLE) AREA', T96,
* 'Z = ', I6 /
* T20, 'TOTAL NUMBER OF PIXELS LESS PIXELS IN EXCLUSION AREA', T92,
* 'Y-Z = F = ', I6 /
* T20, 'NUMBER OF PIXELS CLASSIFIED AS ', A4, ' BEFORE THRESHOLDING',
* T96, 'G = ', I6 /
* T20, 'NUMBER OF PIXELS CLASSIFIED AS OTHER BEFORE THRESHOLDING',
* T96, 'H = ', I6 /
* T20, 'NUMBER OF PIXELS CLASSIFIED AS ', A4, ' AFTER THRESHOLDING',
* T96, 'I = ', I6 /
* T20, 'NUMBER OF PIXELS CLASSIFIED AS OTHER AFTER THRESHOLDING',
* T96, 'J = ', I6 /
530 FORMAT(T20, 'NUMBER OF PIXELS CLASSIFIED AS ', A4, ' WHICH WERE THRES
* HOLD', T92, 'G-I = K = ', I6 /
* T20, 'NUMBER OF PIXELS CLASSIFIED AS OTHER WHICH WERE THRESHOLDED',
* T92, 'H-J = L = ', I6 /
* T20, 'TOTAL NUMBER OF PIXELS THRESHOLDED', T92, 'K+L = M = ', I6 /
540 FORMAT(T20, 'PROPORTION OF ', A4, ' PIXELS THRESHOLDED', T92, 'K/G = N = ',
* F6.3 / T20, 'PROPORTION OF OTHER PIXELS THRESHOLDED', T92, 'L/H = O = ',
* F6.3 / T20, 'PROPORTION OF ', A4, ' PIXELS THRESHOLDED (OF THE TEST SP
* IT)', T92, 'K/F = P = ', F6.3 /
* T20, 'PROPORTION OF OTHER PIXELS THRESHOLDED (OF THE TEST SITE)',
* T92, 'L/F = Q = ', F6.3 /
550 FORMAT(T20, 'PROPORTION OF PIXELS THRESHOLDED', T88, '(K+L)/F = R = ',
* F6.3 / T20, 'PROPORTION OF ', A4, ' BEFORE THRESHOLDING', T92, 'G/F = S = ',
* F6.3 / T20, 'PROPORTION OF OTHER BEFORE THRESHOLDING', T92, 'H/F = T = ',
* F6.3 / T20, 'PROPORTION OF ', A4, ' AFTER THRESHOLDING', T92, 'I/F = U = ',
* F6.3 / T20, 'PROPORTION OF OTHER AFTER THRESHOLDING', T92, 'J/F = V = ',
* F6.3 / T20, 'PROPORTION OF PIXELS TO BE CONSIDERED AS OTHER', T88,
* '(G+K)/F = X = ', F6.3 /
560 FORMAT(T10, '101(' - ') )
570 FORMAT(1H, 'T10, ' - ', T19, ' - ', T31, ' - ', T42, ' - ', T57, ' - ', T72, ' - ', T82,
* ' - ', T94, ' - ', T110, ' - ')
575 FORMAT(T44, 'COMPUTED', T96, 'COMPUTED')
580 FORMAT(T33, 'PROPOR.', T44, 'PROPOR. LESS', T59, 'NO. OF PX.', T74,
* 'PROPOR.', T96, 'PROPOR. LESS')
585 FORMAT(T21, 'NUMBER', T33, 'BEFORE', T44, 'TRUE PROPOR.', T59,
* 'CLSFY. AFTER', T74, 'AFTER', T84, 'PROPOR.', T96, 'TRUE PROPOR.')
590 FORMAT(T12, 'CROP', T21, 'OF PIXELS', T33, 'THRS', T44,
* 'BEFORE THRS', T59, 'THRESHOLD', T74, 'THRS', T84, 'THRS', T96,
* 'AFTER THRS')
595 FORMAT(T25, '(G)', T35, '(S)', T47, '(S-D)', T62, '(I)',
* T76, '(U)', T86, '(P)', T99, '(U-D)')
600 FORMAT(T12, '14, T23, '16, T33, F6.3, T46, F6.3, T61, '16, T74, F6.3, T84, F6.3,
* T98, F6.3)
605 FORMAT(T25, '(H)', T35, '(T)', T47, '(T-E)', T61, '(H+K)',
* T76, '(W)', T86, '(Q)', T99, '(W-E)')
610 FORMAT(/)
611 FORMAT(/)
615 FORMAT(' T12, 'OTHER')
270 FORMAT(/T15, 'PTS. BEFORE', T29, 'PCT. OF', T44, 'PTS. AFTER',
* T58, 'PCT. OF', T70, 'PCT. OF', T87, 'PTS.', T99, 'PCT. OF',
* T111, 'PCT. OF', T17, 'THRES.', T29, 'TOTAL', T46, 'THRES.',
* T58, 'TOTAL', T87, 'THRES.', T99, 'TOTAL')
275 FORMAT(1H, 'T5, 'SUBCLASS', T70, 'SUBCLASS', T111, 'SURCLASS')
276 FORMAT(/)
280 FORMAT(T5, 'A4, T17, '16, T29, F6.2, T46, '16, T58,
* F6.2, T70, F6.2, T87, '16, T99, F6.2, T111, F6.2)
285 FORMAT(/T15, 'PTS. THRESHOLDED IN DISPLAY', 1X, I10 /
* T15, 'PTS. THRESHOLDED IN CLASSIFY', I10 /
* T35, 'TOTAL', 3X, T10, T60, 'PCT. = ', F6.2)
286 FORMAT(1H, 'T5, 'CLASS', T70, 'CLASS', T111, 'CLASS')
287 FORMAT(1H, 'T5, 'CATEGORY', T70, 'CATEGORY', T111, 'CATEGORY')
288 FORMAT(T29, 'CLSF. FLD.', T58, 'CLSF. FLD.', T99, 'CLSF. FLD.',
* T111, 'THRES.')
RETURN
END

```

PRT02380
 PRT02390
 PRT02400
 PRT02410
 PRT02420
 PRT02430
 PRT02440
 PRT02450
 PRT02460
 PRT02470
 PRT02480
 PRT02490
 PRT02500
 PRT02510
 PRT02520
 PRT02530
 PRT02540
 PRT02550
 PRT02560
 PRT02570
 PRT02580
 PRT02590
 PRT02600
 PRT02610
 PRT02620
 PRT02630
 PRT02640
 PRT02650
 PRT02660
 PRT02670
 PRT02680
 PRT02690
 PRT02700
 PRT02710
 PRT02720
 PRT02730
 PRT02740
 PRT02750
 PRT02760
 PRT02770
 PRT02780
 PRT02790
 PRT02800
 PRT02810
 PRT02820
 PRT02830
 PRT02840
 PRT02850
 PRT02860
 PRT02870
 PRT02880
 PRT02890
 PRT02900
 PRT02910
 PRT02920
 PRT02930
 PRT02940
 PRT02950
 PRT02960
 PRT02970
 PRT02980
 PRT02990
 PRT03000
 PRT03010
 PRT03020
 PRT03030
 PRT03040
 PRT03050
 PRT03060
 PRT03070
 PRT03080
 PRT03090
 PRT03100

```

SRRROUTINE REDIF3(TSTSAV,TSTFLD,TSTVER,VDIM,
* GTUNIT,GTFILE,AIUNIT,AIFILE,PPUNIT,PPFILE,
* NAMECT,ALP,DESSAV,DESFLD,DESV,NOFLD4)
IMPLICIT INTEGER (A-H,O-Z)

*** CODE ADDED TO INCLUDE LIST PROCESSING

REAL ALP(2)
-----
PURPOSE.. READS AND ANALYZES SUPERVISOR CONTROL CARDS
FOR ' DISPLAY'
-----

EQUIVALENCE (HED1(1),HEAD(4)), (DATE(1),HEAD(22)),
1 (HED2(1),HEAD(30)), (COMENT(1),HEAD(48))
-----
INCLUDE COMRK6
INCLUDE CMRK10,LIST
COMMON/GLOBAL/HEAD(63),MAPTAP,DATEPA,SAVTAP,RMFILE,BMKEY,
* HISFIL,HISKEY,TRFORM,ERPTP,ERPKEY,MAPUNT,NOFILE,
* DRUMAD,DRMKOS,PAGSIZ,DATFIL,STAFIL,ASAV,ASAVFL
* ,NHSTUN,NHSTFI,STRUN,MAPFIL
* ,DOTUNT,DOTFIL,NCHPAS,TRNSFL,BMTRFL,HISTFL,PCHUNT,
* CRDUNT,PRUNT,RANDIO
COMMON/DISPL/CATFLG,CATNAM(61),CLSNAM(61),SUBNAM(61),SUBNO(60),
* SUBCAT(60),CLSSUB(60),NOMAP,TOTVT3,NOSUB3,
* PCFKEY,TSTKEY,TRNKEY,THRSKY,STATKY,EMPTRS,THRSVA,
* PLTKEY,BMFLG,RMCOMB,RMFEAT,CDATE(2),
* FLDSV2,FIELD2,VERTX2,FLDSV3,FIELD3,VERTX3,PCTID3,
* THRS(60),SYMMTX(66),HIGH(60),CON(60)
* ,FLNKEY,NOFLD2,NOFLD3,NOFET2,FETVC2(30)
* ,NOSUR2,NOTREFD,TOTVT2,NOCLS2
* ,KATNO(60),NOCAT,FILTER,MAPFMT
* ,DESKEY,DESUNI,DFSOTH,CHOP,ACROP,AOTHER,ATOTAL
* ,SITE(6),ANALYS(5),CAM(15),CHPKEY,KEPPTS(60)
* ,DOTKEY,DOTERR
REAL HIGH,CHIN
REAL ACROP,AOTHER,ATOTAL,X
REAL THRES
DIMENSION DESSAV(4,50),DESFLD(5,50),DESV(1100)
DIMENSION TSTSAV(4,200),TSTFLD(5,200),TSTVER(VDIM),
* OPT(20),COMVEC(2),INF(7),ACARD(20)
3,CODBCD(10),EQUCOM(3),CARD2(62),CARD1(80),SLASH(2)
4,COMENT(15),DATE(3),HED1(15),HED2(15)

DATA OPT/'SYMB',SITE,'OPT1','THRE','ANAL',
* COMM,'HED1',HED2,'DATE','PROC','ACRE',
* ,END,'FORM','CROP',
* ,GTUN,'AIUN','PPUN','NAME','ALPH',
DATA OPTNUM/12,ENDBCD/'SEND',THSRC/'THRS',BLANK/'',
1 COMMA/','/,'CODBCD/','S','O','P','N','E','C','T','F',
2,EQUCOM/2,'',COMVEC/1,'',SLASH/1,'/'

DATA LO/'O',LI/'I',LF/'F',LU/'U',
DATA LL/'L',LS/'S',LT/'T',LC/'C',LN/'N',OTHER/'OTHE',
* UNIDEN/'UNID'
DIMENSION SYMMT(66),EQUVEC(2)
DATA SYMMT/1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49,50,51,52,53,54,55,56,57,58,59,60,61,62,63,64,65,66,67,68,69,70,71,72,73,74,75,76,77,78,79,80,81,82,83,84,85,86,87,88,89,90,91,92,93,94,95,96,97,98,99,100,101,102,103,104,105,106,107,108,109,110,111,112,113,114,115,116,117,118,119,120,121,122,123,124,125,126,127,128,129,130,131,132,133,134,135,136,137,138,139,140,141,142,143,144,145,146,147,148,149,150,151,152,153,154,155,156,157,158,159,160,161,162,163,164,165,166,167,168,169,170,171,172,173,174,175,176,177,178,179,180,181,182,183,184,185,186,187,188,189,190,191,192,193,194,195,196,197,198,199,200,201,202,203,204,205,206,207,208,209,210,211,212,213,214,215,216,217,218,219,220,221,222,223,224,225,226,227,228,229,230,231,232,233,234,235,236,237,238,239,240,241,242,243,244,245,246,247,248,249,250,251,252,253,254,255,256,257,258,259,260,261,262,263,264,265,266,267,268,269,270,271,272,273,274,275,276,277,278,279,280,281,282,283,284,285,286,287,288,289,290,291,292,293,294,295,296,297,298,299,300,301,302,303,304,305,306,307,308,309,310,311,312,313,314,315,316,317,318,319,320,321,322,323,324,325,326,327,328,329,330,331,332,333,334,335,336,337,338,339,340,341,342,343,344,345,346,347,348,349,350,351,352,353,354,355,356,357,358,359,360,361,362,363,364,365,366,367,368,369,370,371,372,373,374,375,376,377,378,379,380,381,382,383,384,385,386,387,388,389,390,391,392,393,394,395,396,397,398,399,400,401,402,403,404,405,406,407,408,409,410,411,412,413,414,415,416,417,418,419,420,421,422,423,424,425,426,427,428,429,430,431,432,433,434,435,436,437,438,439,440,441,442,443,444,445,446,447,448,449,450,451,452,453,454,455,456,457,458,459,460,461,462,463,464,465,466,467,468,469,470,471,472,473,474,475,476,477,478,479,480,481,482,483,484,485,486,487,488,489,490,491,492,493,494,495,496,497,498,499,500,501,502,503,504,505,506,507,508,509,510,511,512,513,514,515,516,517,518,519,520,521,522,523,524,525,526,527,528,529,530,531,532,533,534,535,536,537,538,539,540,541,542,543,544,545,546,547,548,549,550,551,552,553,554,555,556,557,558,559,560,561,562,563,564,565,566,567,568,569,570,571,572,573,574,575,576,577,578,579,580,581,582,583,584,585,586,587,588,589,590,591,592,593,594,595,596,597,598,599,600,601,602,603,604,605,606,607,608,609,610,611,612,613,614,615,616,617,618,619,620,621,622,623,624,625,626,627,628,629,630,631,632,633,634,635,636,637,638,639,640,641,642,643,644,645,646,647,648,649,650,651,652,653,654,655,656,657,658,659,660,661,662,663,664,665,666,667,668,669,670,671,672,673,674,675,676,677,678,679,680,681,682,683,684,685,686,687,688,689,690,691,692,693,694,695,696,697,698,699,700,701,702,703,704,705,706,707,708,709,710,711,712,713,714,715,716,717,718,719,720,721,722,723,724,725,726,727,728,729,730,731,732,733,734,735,736,737,738,739,740,741,742,743,744,745,746,747,748,749,750,751,752,753,754,755,756,757,758,759,760,761,762,763,764,765,766,767,768,769,770,771,772,773,774,775,776,777,778,779,780,781,782,783,784,785,786,787,788,789,790,791,792,793,794,795,796,797,798,799,800,801,802,803,804,
```

FILE REDIF3

```

LOGICAL*1 LM(4),LCROP(4)
EQUIVALENCE (CARD2(1),CARD1(11)),(M,LM(1)),(LCROP(1),KCROP)
DO 5 I=1,66
5 SYMMTX(I)=SYMMT(I)
-----
INIZ
-----
C*** CODE ADDED NOV 13, 1978 TO INCLUDE LIST PROCESSING
      GTUNIT = 0
      GTFILF = 0
      AIUNIT = 0
      AIFILE = 0
      PPUNIT = 0
      PPFILE = 0
      NAMECT = BLANK
      ALP(1) = 0.
      ALP(2) = 0.
DOTERR=0
DOTKEY = 0
TSTKEY = 0
TRNKEY = 0
STATKY = 0
PCFOKY = 0
PLTKEY = 0
MAPFMT = 0
NOMAP=1
FILTER=0
THRSVA = 0
FMPTRS = 0
SYMCNT = 0
THRSKY=0
STOPFG = 0
GRPTR = 0
NOGRP = 0
NOTRFD = 0
TYPE=0
CRPKEY=0
THSCNT=0
DESKEY=0
TSTCNT = 0
LOGSWT = 0
PCCLKY = 1
SYMMAX = 60
DO 6 I=1,SYMMAX
THRES(I) = 0
6 CONTINUE
NOFLO3=200

READ IN SUPERVISOR CARDS
-----
SET UP REREAD BUFFER
RRUNIT=30
CALL REREAD(RRUNIT,80)

NOW READ THE CARD INTO THE BUFFER
10 CONTINUE
15 READ(21,180) (ACARD(I),I=1,20)
180 FORMAT(20A4)
WRITE(RRUNIT,180) (ACARD(I),I=1,20)
REWIND RRUNIT
READ(30,104) CODE,CARD2
104 FORMAT(A4,6X,62A1)
REWIND RRUNIT
17 WRITE(6,304) CODE,CARD2
304 FORMAT(I7,A4,6X,62A1)
19 COL = 0
```

RED00770
RED00780
RED00790
RED00800
RED00810
RED00820
RED00830
RED00840
RED00850
RED00860
RED00870
RED00880
RED00890
RED00900
RED00910
RED00920
RED00930
RED00940
RED00950
RED00960
RED00970
RED00980
RED00990
RED01000
RED01010
RED01020
RED01030
RED01040
RED01050
RED01060
RED01070
RED01080
RED01090
RED01100
RED01110
RED01120
RED01130
RED01140
RED01150
RED01160
RED01170
RED01180
RED01190
RED01200
RED01210
RED01220
RED01230
RED01240
RED01250
RED01260
RED01270
RED01280
RED01290
RED01300
RED01310
RED01320
RED01330
RED01340
RED01350
RED01360
RED01370
RED01380
RED01390
RED01400
RED01410
RED01420
RED01430
RED01440
RED01450
RED01460
RED01470
RED01480
RED01490
RED01500
RED01510
RED01520

ORIGINAL PAGE IS
OF POOR QUALITY

FILE REDIF3

```

DO 20 I=1,20
  IF (OPT(1).EQ.CODE) GO TO (100,200,300,400,600,710,720,730,
    * 740,760,770,800,750,780,210,211,212,230,240), I
20  CONTINUE
    GOTO 1500
C
C
C
  GET SYMROLS
  -----
100 IF ( SYMCNT .GE. SYMMAX ) GOTO 10
    SYMCNT = SYMCNT+1
    SYMMTX(SYMCNT) = BLANK
    M = NXTCHR(CARD2,COL)
    IF ( M .EQ. BLANK ) GO TO 10
    IF ( M .EQ. COMMA ) GO TO 100
    SYMMTX(SYMCNT) = CARD2(COL)
110 M = NXTCHR(CARD2,COL)
    IF ( M .EQ. BLANK ) GO TO 10
    IF ( M .NE. COMMA ) GO TO 110
    GO TO 100
C*
C*
C*  SITE NAME
200 READ(30,201) SITE
201 FORMAT(10X,6A4)
    REWIND RRUNIT
    GO TO 10
C
C*** CODE ADDED NOV 13,1978. TO INCLUDE LIST PROCESSING
C*** READ GT AI OR PP UNIT AND FILE NUMBERS
210 IPAT = 16
    GO TO 214
211 IPAT = 17
    GO TO 214
212 IPAT = 18
214 M = NXTCHR(CARD2,COL)
    IF (M.EQ.BLANK) GO TO 216
    IF (M.NE.LU) GO TO 215
    M = FIND12(CARD2,COL,EQUVEC)
    IF (M.NE.2) GO TO 216
    ISTART = 0
    M = NUMBER(CARD2,COL,IPATT,ISTART)
    M = FIND12(CARD2,COL,EQUVEC)
    IF (M.NE.2) GO TO 216
    ISTART = 0
    M = NUMBER(CARD2,COL,IPATTT,ISTART)
    GO TO 218
215 IF (M.NE.LF) GO TO 216
    M = FIND12(CARD2,COL,EQUVEC)
    IF (M.NE.2) GO TO 216
    ISTART = 0
    M = NUMBER(CARD2,COL,IPATTT,ISTART)
    M = FIND12(CARD2,COL,EQUVEC)
    IF (M.NE.2) GO TO 216
    ISTART = 0
    M = NUMBER(CARD2,COL,IPATT,ISTART)
    GO TO 218
216 WRITE(6,217) OPT(IPAT)
217 FORMAT(' ERROR ON ',A4,' CONTROL CARD ')
    GO TO 10
218 IF (IPAT.NE.16) GO TO 219
    GTUNIT = IPATT
    GTFILF = IPATTT
    GO TO 221
219 IF (IPAT.NE.17) GO TO 220
    AIUNIT = IPATT
    AIFILF = IPATTT
    GO TO 221
220 PPUNIT = IPATT
    PPFILE = IPATTT
221 DOTKEY = 1
    GO TO 10
C*** SELECTED LIST CLASS NAME
230 NAMECT = NXTCHR(CARD2,COL)
    IF (NAMECT.NE.BLANK) GO TO 10

```

RED01530
 RED01540
 RED01550
 RED01560
 RED01570
 RED01580
 RED01590
 RED01600
 RED01610
 RED01620
 RED01630
 RED01640
 RED01650
 RED01660
 RED01670
 RED01680
 RED01690
 RED01700
 RED01710
 RED01720
 RED01730
 RED01740
 RED01750
 RED01760
 RED01770
 RED01780
 RED01790
 RED01800
 RED01810
 RED01820
 RED01830
 RED01840
 RED01850
 RED01860
 RED01870
 RED01880
 RED01890
 RED01900
 RED01910
 RED01920
 RED01930
 RED01940
 RED01950
 RED01960
 RED01970
 RED01980
 RED01990
 RED02000
 RED02010
 RED02020
 RED02030
 RED02040
 RED02050
 RED02060
 RED02070
 RED02080
 RED02090
 RED02100
 RED02110
 RED02120
 RED02130
 RED02140
 RED02150
 RED02160
 RED02170
 RED02180
 RED02190
 RED02200
 RED02210
 RED02220
 RED02230
 RED02240
 RED02250
 RED02260
 RED02270
 RED02280

FILE REDIF3

```

399 COL=COL+1
M=NXTCHR(CARD2,COL)
IF(M.EQ.LL)GO TO 3990
IF(M.EQ.LS)GO TO 3991
GO TO 315
3990 FILTER=1
GO TO 390
C-
3991 IX=0
DO 3995 I=1,NOSUR2
IF(KEPPTS(I).GT.NOFET2)GO TO 3995
IF(IX.NE.0)GO TO 3993
WRITE(6,3992)
3992 FORMAT(1H0.////)
IX=1
3993 WRITE(6,3994) I , KEPPTS(I) , NOFET2
3994 FORMAT(' ***** FISHER THRESHOLD REQUESTED-NOT PERFORMED',/,BX,
1'... NO. SAMPLES FOR SURCLASS',I6,'(=',I6,') IS LESS THAN OR EQUAL
2 TO NUMBER OF CHANNELS (=',I6,')',/,)
3995 CONTINUE
IF(IX.EQ.0)GO TO 3996
WRITE(6,3992)
GO TO 390
C-
C- -- SET THRESHOLD KEYS FOR FISHER THRESHOLDING,TURN OTHERS OFF
3996 EMPTRS=0
THRSVA=0
THRSKY=4
C-
C- GO TO 390
C-
C- READ IN THRESHOLDS
C-
C- -----
400 CONTINUE
I = SYMMAX-THSCNT
THSCNT = THSCNT+FLTNUM(CARD2,COL,THRES(THSCNT+1),I)
GO TO 10
C-
C- ANALYST NAME
C-
600 READ(30,601)ANALYS
601 FORMAT(10X,5A4)
REWIND RRUNIT
GO TO 10
C-
C- COMMENT
C-
C- -----
710 READ(30,999A) COMENT
999A FORMAT(10X,15A4)
REWIND RRUNIT
GOTO 10
C-
C- HED1
C-
C- -----
720 READ(30,999B) HED1
REWIND RRUNIT
GOTO 10
C-
C- HED2
C-
C- -----
730 READ(30,999A) HED2
REWIND RRUNIT
GOTO 10
C-
C- DATE
C-
C- -----
740 READ(30,999A) DATE
REWIND RRUNIT
GO TO 10

```

RED03050
 RED03060
 RED03070
 RED03080
 RED03090
 RED03100
 RED03110
 RED03120
 RED03130
 RED03140
 RED03150
 RED03160
 RED03170
 RED03180
 RED03190
 RED03200
 RED03210
 RED03220
 RED03230
 RED03240
 RED03250
 RED03260
 RED03270
 RED03280
 RED03290
 RED03300
 RED03310
 RED03320
 RED03330
 RED03340
 RED03350
 RED03360
 RED03370
 RED03380
 RED03390
 RED03400
 RED03410
 RED03420
 RED03430
 RED03440
 RED03450
 RED03460
 RED03470
 RED03480
 RED03490
 RED03500
 RED03510
 RED03520
 RED03530
 RED03540
 RED03550
 RED03560
 RED03570
 RED03580
 RED03590
 RED03600
 RED03610
 RED03620
 RED03630
 RED03640
 RED03650
 RED03660
 RED03670
 RED03680
 RED03690
 RED03700
 RED03710
 RED03720
 RED03730
 RED03740
 RED03750
 RED03760
 RED03770
 RED03780
 RED03790
 RED03800

3

750

33

CCC

ORIGINAL PAGE IS
OF POOR QUALITY

33

6

CCC

6

C4
C4

333

C4

cc

C-

6.

2

88

RE003810
RE003820
RE003830
RE003840
RE003850
RE003860
RE003870
RE003880
RE003890
RE003900
RE003910
RE003920
RE003930
RE003940
RE003950
RE003960
RE003970
RE003980
RE003990
RE004000
RE004010
RE004020
RE004030
RE004040
RE004050
RE004060
RE004070
RE004080
RE004090
RE004100
RE004110
RE004120
RE004130
RE004140
RE004150
RE004160
RE004170
RE004180
RE004190
RE004200
RE004210
RE004220
RE004230
RE004240
RE004250
RE004260
RE004270
RE004280
RE004290
RE004300
RE004310
RE004320
RE004330
RE004340
RE004350
RE004360
RE004370
RE004380
RE004390
RE004400
RE004410
RE004420
RE004430
RE004440
RE004450
RE004460
RE004470
OPEDU4480
RE004490
RE004500
RE004510
RE004520
RE004530
RE004540
RE004550
RE004560

FILE REDIF3

```

C*
TOTVT3=0
NOFLD3=1
IPT=1
NOFLD4=1
PPT=1
840 ICK=LAREAD(TSTSAV(1,NOFLD3),TSTVER(IPT),INF,NV)
IF(ICK.EQ.-3)GO TO 865
IF(ICK.EQ.-2)GO TO 850
IF(ICK.EQ.-1)GO TO 860
IF(ICK.EQ.0)GO TO 870
TSTSAV(2,NOFLD3)=CLSIND
TSTSAV(3,NOFLD3)=SUBIND
TSTSAV(4,NOFLD3)=NV
TSTFLD(1,NOFLD3)=INF(1)
TSTFLD(2,NOFLD3)=INF(2)
TSTFLD(3,NOFLD3)=INF(4)
TSTFLD(4,NOFLD3)=INF(5)
TSTFLD(5,NOFLD3)=IPT
IPT=IPT + 2*NV
NOFLD3 = NOFLD3 + 1
TOTVT3=TOTVT3+NV
GO TO 840
C* SURCLASS NAME
C*
850 READ(30,851)NAME
REWIND RRUNIT
851 FORMAT(10X,A4)
DO 852 I=1,NOSUH2
IF(NAME.EQ.SUBNAM(I))GO TO 854
852 CONTINUE
WRITE(6,853)NAME
853 FORMAT(1 *ERROR ON SUBCLASS NAME CARD -',A4,' DOES NOT MATCH A SUR
*CLASS FROM THE MAPTAP FILE *')
CALL CMERR
854 SUBIND=I
CLSIND=CLSSUB(I)
GO TO 840
C*
C* CLASSNAME CARD
C*
860 READ(30,851)NAME
REWIND RRUNIT
DO 861 I=1,NOCLS2
IF(NAME.EQ.CLSNAM(I))GO TO 863
861 CONTINUE
WRITE(6,862)NAME
862 FORMAT(1 *ERROR ON CLASSNAME CARD -',A4,' DOES NOT MATCH A CLASS NR
*AME FROM THE MAPTAP FILE *')
CALL CMERR
863 CLSIND=I
SUBIND=0
GO TO 840
C*
C* DESIGNATED FIELDS
C*
865 READ(30,851)TEST
REWIND RRUNIT
SUBIND=1
CLSIND = NOSUH3 + 4
IF(TEST.EQ.OTHER)SUBIND=2
IF(TEST.EQ.OTHER)CLSIND=NOSUH3+5
IF(TEST.NE.OTHER.AND.TEST.NE.UNIDEN)GO TO 8066
8065 ICK=LAREAD(DESSAV(1,NOFLD4),DESVER(PPT),INF,NV)
IF(ICK.EQ.-3)GO TO 865
IF(ICK.EQ.-2)GO TO 850
IF(ICK.EQ.-1)GO TO 860
IF(ICK.EQ.0)GO TO 870
DESSAV(2,NOFLD4)=CLSIND
DESSAV(3,NOFLD4)=SUBIND
DESSAV(4,NOFLD4)=NV
DESFLO(1,NOFLD4)=INF(1)
DESFLO(2,NOFLD4)=INF(2)
DESFLO(3,NOFLD4)=INF(4)
DESFLO(4,NOFLD4)=INF(5)
DESFLO(5,NOFLD4)=PPT

```

RED04570
 RED04580
 RED04590
 RED04600
 RED04610
 RED04620
 RED04630
 RED04640
 RED04650
 RED04660
 RED04670
 RED04680
 RED04690
 RED04700
 RED04710
 RED04720
 RED04730
 RED04740
 RED04750
 RED04760
 RED04770
 RED04780
 RED04790
 RED04800
 RED04810
 RED04820
 RED04830
 RED04840
 RED04850
 RED04860
 RED04870
 RED04880
 RED04890
 RED04900
 RED04910
 RED04920
 RED04930
 RED04940
 RED04950
 RED04960
 RED04970
 RED04980
 RED04990
 RED05000
 RED05010
 RED05020
 RED05030
 RED05040
 RED05050
 RED05060
 RED05070
 RED05080
 RED05090
 RED05100
 RED05110
 RED05120
 RED05130
 RED05140
 RED05150
 RED05160
 RED05170
 RED05180
 RED05190
 RED05200
 RED05210
 RED05220
 RED05230
 RED05240
 RED05250
 RED05260
 RED05270
 RED05280
 RED05290
 RED05300
 RED05310
 RED05320

```
PPT=3PT-2*NV  
NOFLD4=NOFLD4+1  
GO TO 8065
```

C***
C***
C*** THIS CODE ADDED AUG 31, 1978 TO ALLOW CLASSNAME TO APPEAR
ON DESIGNATED CARD THIS FORCES RECLASSIFICATION OF
DESIGNATED PIXELS INTO THE FIRST SURCLASS ASSIGNED
TO THE CLASS NAMED ON THE DESIG CARD STARTING AT COL 11
C***
C***
C***

8066 DO 866 I = 1,60
IF (TEST.NE.CLSNAM(I)) GO TO 866
I = I + 1
GO TO 867
CONTINUE
866 WRITE(PRTUNT,4005) TEST
4005 FORMAT(' DESIGNATED FIELD OF CLASSNAME',A4,'DOES NOT
MATCH A CLASSNAME ON MAPTAP---DEFAULTING TO UNIDEN')
GO TO 8065
867 DO 868 I = 1,60
IF (II.NE.CLSSUB(I)) GO TO 868
SUBIND = 0
CLSIND = I
GO TO 869
CONTINUE
868 WRITE(PRTUNT,4006) TEST
4006 FORMAT(' DESIGNATED FIELD',A4,'CANNOT BE MATCHED
DEFAULTING TO UNIDENTIFIABLE')
869 GO TO 8065

C**
C** SEND* - END OF TEST OR DESIGNATED FIELDS
C**
870 NOFLD3 = NOFLD3-1
NOFLD4=NOFLD4-1
IF(NOFLD3.GT.0)TSTKEY=1

C*** CODE ADDED NOV 13, 1978 TO INCLUDE LIST PROCESSING
C***
IF (DOTKEY.EQ.0) GO TO 900
TRNKEY = 0
900 CONTINUE
IF(NOFLD4.GT.0)DESKEY=1

C**
C** SET THRESHOLD AND OUTLINE SYMMROLS
C**
SYMMTX(NOSUR3)=THRSYM
SYMMTX(NOSUR3+1)=TRNSYM
SYMMTX(NOSUR3+2)=TSTSVM
SYMMTX(NOSUR3+3)=DUPSYM
SYMMTX(NOSUR3+4)=DESSYM
SYMMTX(NOSUR3+5)=DESSYM

C***
C*** GO HOME

RETURN
ERROR ROUTINES

1500 WRITE (6,15002) CODE, CARD?
15002 FORMAT(/IX,A4,6X,62A1/' INVALID CONTROL CARD-CHECK SPELLING OF KEY
WORD')
GO TO 10
END

ORIGINAL PAGE IS
OF POOR QUALITY

ORIGINAL PAGE IS
OF POOR QUALITY

FILE: RNORM FORTRAN A

```

FUNCTION RNORM(X)
  DIMENSION A(7)
  DATA A / .43063E-4, .276567E-3, .1520143E-3,
    .9270527E-2, .422820E-1, .70523E-1, 1.0 /
  Y=ABS(X)/1.414213
  RNORM=0.
  DO 1 I=1,7
    RNORM=RNORM*Y*A(I)
  CALL OVERFL(INDCT)
  IF(INDCT.NE.2) GO TO 3
1 CONTINUE
  RNORM=.5*(((1./RNORM)**2)**2)**2)**2)
2 IF(X.GT.0.0) RNORM = 1.0-RNORM
  RETURN
3 RNORM=0.0
  GO TO 2
END

```

```

RN000010
RN000020
RN000030
RN000040
RN000050
RN000060
RN000070
RN000080
RN000090
RN000100
RN000110
RN000120
RN000130
RN000140
RN000150
RN000160
RN000170

```

ORIGINAL PAGE IS
OF POOR QUALITY

FILE SETUP3

```

      SUBROUTINE SETUP3(ARRAY, TOP, GTUNIT, GTFILE,
      *   AIUNIT, AIFILE, PPUNIT, PPFILE, NAMECT, ALP, DESSAV,
      *   DESFLD, DESVER, NOFLD4, STOP)
      IMPLICIT INTEGER (A-Z)

C*** CODE ADDED NOV 13, 1978 TO INCLUDE LIST PROCESSING
      REAL ALP(2)
      LOGICAL OKAY
      INCLUDE CUMHKA, LIST
      INCLUDE CMHKL0, LIST
      COMMON/GLOBAL/HEAD(63), MAPTAP, DATAPE, SAVTAP, BMFILE, RMKEY,
      *   HISFIL, HISKEY, THFORM, ERIP, EHPKEY, MAPUNT, NOFILE,
      *   DRUMAD, DRUMDS, PAGESZ, DATAFIL, STAFIL, ASAV, ASAVFL,
      *   NHSTUN, NHSTFI, SCTRUN, MAPFIL,
      *   DOTUNT, DOTFIL, NCHPAS, TRNSFL, BMTRFL, HISTFL, PCHUNT,
      *   CRDUNT, PHTUNT, HANDIO
      COMMON/DISPL/CATLG, CATNAM(61), CLSNAM(61), SUHNAME(61), SUBNO(60),
      *   SUHCAT(60), CLSSUH(60), NOMAP, TOTVT3, NOSUB3,
      *   PCHKEY, TSTKEY, TRNKEY, THRSKY, STATKY, EMPTRS, THRSVA,
      *   PLKEY, HMFLG, HMCOMR, RMFFAT, CDATE(2),
      *   FLDV2, FIELD2, VERTX2, FLDV3, FIELD3, VERTX3, PCTID3,
      *   THRS(60), SYMTX(66), HIGH(60), CON(60),
      *   FLDKEY, NOFLD2, NOFLD3, NOFT2, FETVC2(30),
      *   NOSUP2, NOTREFD, TOTVT2, NOCLS2,
      *   KATNO(60), NOCAT, FILTER, MAPFMT,
      *   DESKEY, DESUNI, DESOTH, CROP, ACROP, AOTHER, ATOTAL,
      *   SITE(6), ANALYS(5), CAM(15), CRPKEY, KEPPIS(60),
      *   DOTKEY, DOTERR

CSEND
      DATA A//A//, B//NO//
      DATA BLANK//, //

-----
      PURPOSE.. LOCATES FILE ON 'MAPTAP' AND COORDINATES
      ROUTINES TO ANALYSE 'DISPLAY' CONTROL CARDS
      I
-----

      DIMENSION NUMVEC(30), FILVEC(2)
      DIMENSION DESSAV(4,50), DESFLD(5,50), DESVER(1100)
      DIMENSION CARD(62), ARRAY(1)

C*
C*   DIMENSION DOTCAT(62)
C*
C*   EQUIVALENCE ( DOTCAT(1) , CARD(1) )
C*
C*   DATA YBCD/ 'Y' /, NHCD/ 'N' /, FILVEC/ 1 , 'F' /
C*
C*   DIMENSION EQUVEC(2)
      DATA DRUM//DRUM//
      DATA UHCD//U//, FBOD//F//, IHCD//I//, EQUVEC/1, 'E'/
      DIMENSION SLASH(2)
      DATA SLASH/1, '/'

C*
C*   INIZ
      -----

      READ FIRST CONTROL CARD FOR MAPTAP UNIT AND FILE NUMBER

      1 READ(CRDUNT,1) CARD
      2 FORMAT(10X,62A1)
      NFILE = 1
      COL = 0
      J = NATCHR(CARD, COL)
      IF (J.EQ.BLANK) GO TO 6
      IF (J.NE.UHCD) GO TO 3
      J = FIND12(CARD, COL, EQUVEC)

```

FILE SETUP3

```

IF (J.NE.2) GO TO 6
ISTART = 0
J = NUMBER(CARD,COL,MAPTAP,ISTART)
J = FIND12(CARD,COL,EQUVEC)
IF (J.NE.2) GO TO 6
ISTART = 0
J = NUMBER(CARD,COL,NFILE,ISTART)
GO TO 10
3 IF (J.NE.IRCD) GO TO 4
J = FIND12(CARD,COL,SLASH)
IF (J.NE.2) GO TO 6
J = MATCHR(CARD,COL)
IF (J.EQ.FRCD) GO TO 5
IF (J.EQ.UBCD) GO TO 2
GO TO 6
4 IF (J.NE.FRCD) GO TO 6
5 J = FIND12(CARD,COL,EQUVEC)
IF (J.NE.2) GO TO 6
ISTART = 0
J = NUMBER(CARD,COL,NFILE,ISTART)
J = FIND12(CARD,COL,EQUVEC)
IF (J.NE.2) GO TO 6
ISTART = 0
J = NUMBER(CARD,COL,MAPTAP,ISTART)
GO TO 10
6 WRITE(6,7)
7 FORMAT(1, ' ERROR ON MAPTAP CARD')
10 CONTINUE
IF (NFILE.LE.0) NFILE = 1
SERIAL = NFILE
C
C GET TAPE READY
40 REWIND MAPTAP
IF (NFILE.EQ.1) GO TO 50
NF = NFILE - 1
CALL FSHSFL(MAPTAP,NF,ISTAT)
IF (ISTAT.EQ.0) GO TO 50
WRITE(6,45) NF,ISTAT
45 FORMAT(/// 5X, '***** DISPLAY/SETUP3 ... ERROR CONDITION ON ATTEMPT
1PT TO POSITION MAPTAP OVER',A,3X,'FILES',// 5X,
2 '***** FSHSFL STATUS CODE =',I4,3X,' ... ABORTING RUN *****',/1H)
REWIND MAPTAP
CALL CMERR
C
C READ MAPTAP
-----
50 CONTINUE
CATFLG=1
READ(MAPTAP) CDATE(1),CDATE(2),BMFLG,BMCOMH,BMFEAT,NOCLS2,
NOFLD2,NOSUB2,NOFLT2,TOTVT2,NOCAT,VARSZ2,
(FETVC2(I),I=1,NOFLT2)
NCAT=NOCAT
IF (NOCAT.GT.0) GO TO 55
CATFLG=0
NCAT=NOCLS2
C* SET BASE ADDRESSES FOR TRAINING FIELD INFORMATION
55 CONTINUE
NOSUB3=NOSUB2+1
FLOS2=1
VERTX2=FLOS2 + NOFLD2*4
FIELD2=VERTX2 + TOTVT2*2
TOP1=FIELD2 + NOFLD2*5
NV=TOTVT2*2
NF=NOFLD2*4
READ(MAPTAP,END=60) (CATNAM(I),I=1,NCAT), (CLSNAM(I),I=1,NOCLS2),
(SUMNO(I),I=1,NOCLS2), (SUMNAM(I),I=1,NOSUB2),
(ARWAY(FLOS2-1),I=1,NF),
(ARWAY(VERTX2-1),I=1,NV),
(SUCAT(I),I=1,NOSUB2), (CLSSUB(I),I=1,NOSUB2),
(KATNO(I),I=1,NOCLS2), (KFPTS(I),I=1,NOSUB2)
GO TO 65
60 STOP=1
C*

```

```

T000770
T000780
T000790
T000800
T000810
T000820
T000830
T000840
T000850
T000860
T000870
T000880
T000890
T000900
T000910
T000920
T000930
T000940
T000950
T000960
T000970
T000980
T000990
T010000
T010010
T010020
T010030
T010040
T010050
T010060
T010070
T010080
T010090
T010100
T010110
T010120
T010130
T010140
T010150
T010160
T010170
T010180
T010190
T010200
T010210
T010220
T010230
T010240
T010250
T010260
T010270
T010280
T010290
T010300
T010310
T010320
T010330
T010340
T010350
T010360
T010370
T010380
T010390
T010400
T010410
T010420
T010430
T010440
T010450
T010460
T010470
T010480
T010490
T010500
T010510
T010520

```

FILE SETUP3

~~12-53~~
232

FILE SETUP3

```

830      FORMAT(///,5X,' SELECTED CATEGORY NAME FOR LIST IS ',A4) SET02290
831      FORMAT(///,5X,' BIAS CORRECTION ALPHAS ARE ',2F10.6) SET02290
803 FORMAT(T10,'APPLY NO THRESHOLDING') SET02300
804 FORMAT(T10,'APPLY CHI SQUARE THRESHOLDS') SET02310
805 FORMAT(T10,'APPLY EMPIRICAL THRESHOLDS') SET02320
806 FORMAT(T10,'OUTLINE THE TRAINING FIELDS') SET02330
808 FORMAT(T10,'OUTLINE THE TEST FIELDS') SET02340
810 FORMAT(T10,'PRINT OUT THE STATISTICS') SET02350
812 FORMAT(T10,'PRINT THE GROUND TRUTH PERFORMANCE SUMMARIES BY FIELD' SET02360
      ) SET02370
C- SET02380
A17 FORMAT(T10,'APPLY FISHER F-DISTRIBUTION THRESHOLDS') SET02390
C- SET02400
A18 FORMAT(T10,'APPLY USER INPUT THRESHOLD VALUES') SET02410
A19 FORMAT(///) SET02420
A20 FORMAT(T10,' DO NOT DISPLAY A CLASSIFICATION MAP') SET02430
A21 FORMAT(T10,' DISPLAY THE HISTOGRAMS OF THE QUADRATIC FORM FOR ALL S SET02440
      *URCLASSES') SET02450
A22 FORMAT(T10,'PERFORM SPATIAL FILTERING') SET02460
A23 FORMAT(T10,'EXCLUDE PIXELS IN THE DESIGNATED AREAS FROM CLASSIFICA SET02470
      *TION SUMMARIES') SET02480
A24 FORMAT(T10,'PRINT THE INTENSIVE TEST SITE SUMMARY REPORT FOR ',A6) SET02490
A25 FORMAT(T10,'PRINT DOT DATA PERFORMANCE SUMMARIES FOR DOT DATA FRO SET02500
      1M FORTRAN UNIT',I3,' FILE NO. ',I3,' TAPE (OR FILE) ',A4) SET02510
C* SET02520
A26 FORMAT(T10,'OUTLINE THE DOTS ON THE CLASSIFICATION MAP' ) SET02530
C SET02540
82 CONTINUE SET02550
C SET02560
C SET02570
C SET02580
      CALL WRTFLD(ARRAY(FLDSV2),ARRAY(VRTX2),NOFLD2,1,CLSNAM,SUBNAM) SET02590
      IF(NOFLD3.LE.0.AND.NOFLD4.LE.0)GO TO 85 SET02600
      IF(TSTKEY.EQ.1)IK=2 SET02610
      IF(TSTKEY.EQ.1)CALL WRTFLD(ARRAY(FLDSV3),ARRAY(VRTX3),NOFLD3,IK, SET02620
      *CLSNAM,SUBNAM) SET02630
      IF(DESKEY.EQ.1)IK=3 SET02640
      IF(DESKEY.EQ.1)CALL WRTFLD(DESSAV,DESVER,NOFLD4,IK,CLSNAM,SUBNAM) SET02650
      85 CONTINUE SET02660
C* SET02670
C* SET02680
C* SET02690
C*** CODE ADDED NOV 13,1978 TO INCLUDE LIST PROCESSING SET02700
C SET02710
      IF (DOTKEY.EQ.0) GO TO 86 SET02720
C* SET02730
C* SET02740
      MAKE SPACE AVAILABLE IN ARRAY FOR DOT DATA INFORMATION, SET02750
      INCLUDING 1000 SCRATCH LOCATIONS FOR TEMPORARY STORAGE OF DOT SET02760
      DATA RETURNED FROM SUBR. RODOTS. SET02770
C* SET02780
      MOVE THE TEST STORAGE ( TSTSAV,TSTFLD,TSTVER ) TO OVERLAY SET02790
      THE INPUT ( MAPTAP ) TRAINING FIELD STORAGE . SET02800
C* SET02810
C* SET02820
      IF ( NOCAT .LE. 0 ) GO TO 108 SET02830
C* SET02840
87 CONTINUE SET02850
      NTSAV = FIELD3 - FLDSV3 SET02860
      NTFLD = VERTX3 - FIELD3 SET02870
      NTVER = TOTVT3 * 2 SET02880
      NMOVE = NTSAV + NTFLD + NTVER SET02890
      FROMAD = FLDSV3 SET02900
      DO 90 I=1,NMOVE SET02910
      II = I SET02920
      90 FROMAD = FLDSV3 + II SET02930
C* SET02940
C* SET02950
      RESET THE TEST/DESIG FIELDS STORAGE BASE ADDRESSES IN ARRAY SET02960
C* SET02970
      FLDSV3 = 1 SET02980
      FIELD3 = FLDSV3 + NTSAV SET02990
      VERTX3 = FIELD3 + NTFLD SET03000
      TOP1 = VERTX3 + NTVER SET03010
C SET03020
      FLDSV2 = TOP1 SET03030
      FIELD2 = TOP1 + 4 SET03040

```

FILE SETUP3

```

      VERTX2 = TOP1 * 5
      TOP2 = VERTX2 * 500
      NOFLD2 = 209
      NOTPFD = 209
      PCTI03 = TOP2
      RETURN

C*
C*
86  NOTRFD=NOFLD2
    IF (TSTKEY.EQ.1) NOTRFD=NOFLD3
    PCTSZ=NOTRFD*NOSUR2
    IF (PCTSZ.LE.(TOP-TOP2)) GO TO 106
C*  MOVE TEXT FIELD INFO SO NO GAPS IN ARRAY IF STORAGE IS NEEDED
    IF (TSTKEY.EQ.0) GO TO 105
C*  MUST TRAINING FIELD INFO BE KEPT IN CORE
    IAD=FLDSV2-1
    IF (TRNKEY.EQ.1) IAD=TOP1-1
    IRD=FLDSV3-1
    NF=4*NOFLD3
    TIME=0
    99 DO 101 I=1,NF
    101 ARRAY(IAD+I)=ARRAY(IRD+I)
        IF (TIME.GT.0) GO TO 102
        IAD=IAD+NF
        IRD=IRD+NF
        NF=NF-1
        TIME=TIME+1
        GO TO 99
    102 IF (TIME.EQ.2) GO TO 103
        IAD=IAD+NF
        IRD=IRD+NF
        NF=NF-1
        TIME=TIME+1
        GO TO 99
    103 CONTINUE
        FLDSV3=TOP1
        IF (TRNKEY.EQ.0) FLDSV3=1
        FIELD3=FLDSV3 * 4*NOFLD3
        VERTX3=FIELD3 * 5*NOFLD3
        TOP2 = VERTX3 * 2*TOTVT3

C*
105 IF ( PCTSZ .GT. ( TOP-TOP2 ) ) GO TO 508
C*
106 PCTI03 = TOP2
C*  SET UP FIELD ARRAY FOR TRAINING FIELDS
C*
    IF ( DOTERR .GT. 0 ) RETURN
C*
107 IF (TSTKEY.EQ.0.OR.TRNKEY.EQ.1) GO TO 400
401 CONTINUE
C*
C*  SET FLOKEY
C*
C*
C*
    IF (TSTKEY.NE.1) GO TO 340
    FLOKEY=1
    DO 310 I=1,NOFLD3
        NS = FLDSV3-1+3*(I-1)*4
        IF (ARRAY(NS).EQ.0) GO TO 320
    310 CONTINUE
        GO TO 200
    340 FLOKEY=1
        DO 350 I=1,NOFLD2
            NS = FLDSV2-1+3*(I-1)*4
            IF (ARRAY(NS).EQ.0) GO TO 320
    350 CONTINUE
        RETURN
    320 FLOKEY=0
    200 RETURN
    201 DOTKEY = NDCAT
        FLOKEY = 0
        RETURN

C
C  ERROR ROUTINES
C

```

SET03050
 SET03060
 SET03070
 SET03080
 SET03090
 SET03100
 SET03110
 SET03120
 SET03130
 SET03140
 SET03150
 SET03160
 SET03170
 SET03180
 SET03190
 SET03200
 SET03210
 SET03220
 SET03230
 SET03240
 SET03250
 SET03260
 SET03270
 SET03280
 SET03290
 SET03300
 SET03310
 SET03320
 SET03330
 SET03340
 SET03350
 SET03360
 SET03370
 SET03380
 SET03390
 SET03400
 SET03410
 SET03420
 SET03430
 SET03440
 SET03450
 SET03460
 SET03470
 SET03480
 SET03490
 SET03500
 SET03510
 SET03520
 SET03530
 SET03540
 SET03550
 SET03560
 SET03570
 SET03580
 SET03590
 SET03600
 SET03610
 SET03620
 SET03630
 SET03640
 SET03650
 SET03660
 SET03670
 SET03680
 SET03690
 SET03700
 SET03710
 SET03720
 SET03730
 SET03740
 SET03750
 SET03760
 SET03770
 SET03780
 SET03790
 SET03800

ORIGINAL PAGE IS
 OF POOR QUALITY

FILE SETUP3

```

C
508 WRITE (6,5044) DIFF
5044 FORMAT (////5X,***** DISPLAY/SETUP3 - CORE OVERFLOW (TOP-TOP2)
1 BY:16,!-- EXECUTION TERMINATED *****/IH1)
CALL CMERR
108 WRITE (6,109)
109 FORMAT (////5X,**** CLASSIFICATION BY CATEGORY (ON MAPTAP ) IS
1 REQUIRED IN ORDER TO PROCESS THE DOT DATA **** / 5X, **** DOT PE
2RFORMANCE SUMMARIES WILL NOT BE OUTPUT **** //// )
C*
GO TO 112
C*
ERROR RETURN - MAX. NO. OF DOTS EXCEEDED
C*
ERROR IN DOTFILE - RESET DOTKEY AND TRNKEY --
C*
112 DOTKEY = 0
TRNKEY=0
RETURN
C*
-----
C*
INTERNAL ROUTINE TO FIND RECTANGULAR COORDINTES FOR TRAINING FIELD
C*
400 CONTINUE
IR=1
IPT=1
DO 20 I=1,NOFLD2
SAMSTR = 100000
SAMEND = 0
LINSTR = 100000
LINEND = 0
NS = FLDSV2-1+4*(I-1)*4
NV = ARRAY(NS)
NS = FIELD2-1+5*(I-1)*5
ARRAY(NS) = IPT
IPT=IPT+NV*2
IE=IR+NV-1
DO 410 J=IR,IE
NS = VERTX2-1+1*(J-1)*2
SAMSTR = MIN0(SAMSTR,ARRAY(NS))
SAMEND = MAX0(SAMEND,ARRAY(NS))
NS = VERTX2-1+2*(J-1)*2
LINSTR=MIN0(LINSTR,ARRAY(NS))
LINEND=MAX0(LINEND,ARRAY(NS))
410 CONTINUE
NS=FIELD2-1+1*(I-1)*5
ARRAY(NS)=LINSTR
NS=FIELD2-1+2*(I-1)*5
ARRAY(NS)=LINEND
NS=FIELD2-1+3*(I-1)*5
ARRAY(NS)=SAMSTR
NS=FIELD2-1+4*(I-1)*5.
ARRAY(NS)=SAMEND
IR=IE+1
20 CONTINUE
GO TO 401
C*
END

```

```

SET03810
SET03820
SET03830
SET03840
SET03850
SET03860
SET03870
SET03880
SET03890
SET03900
SET03910
SET03920
SET03930
SET03940
SET03950
SET03960
SET03970
SET03980
SET03990
SET04000
SET04010
SET04020
SET04030
SET04040
SET04050
SET04060
SET04070
SET04080
SET04090
SET04100
SET04110
SET04120
SET04130
SET04140
SET04150
SET04160
SET04170
SET04180
SET04190
SET04200
SET04210
SET04220
SET04230
SET04240
SET04250
SET04260
SET04270
SET04280
SET04290
SET04300
SET04310
SET04320
SET04330
SET04340
SET04350
SET04360
SET04370
SET04380
SET04390
SET04400
SET04410
SET04420
SET04430

```

FILE: TINORM

| | |
|---|----------|
| FUNCTION TINORM(ALPHA,IFLAG) | TIN00010 |
| DIMENSION A(3),B(3) | TIN00020 |
| DATA A/.010324,.802853,2.515517/,B/.0010308, | TIN00030 |
| 1 .189249,1.432788/ | TIN00040 |
| C----- | TIN00050 |
| C APPROXIMATION TO INVERSE NORMAL DISTRIBUTION | TIN00060 |
| C----- | TIN00070 |
| IF(.NOT.(ALPHA.GT.0..AND.ALPHA.LT.1.)) IFLAG=1 | TIN00080 |
| X=ALPHA | TIN00090 |
| IF(X.GT..5) X=1.-X | TIN00100 |
| Y=SQRT(-2.*ALOG(X)) | TIN00110 |
| TINORM=X-(A(3)+X*(A(2)+X*A(1)))/(1.+X*(B(3)+X*(B(2)+X*B(1)))) | TIN00120 |
| CALL OVRFL(I) | TIN00130 |
| IF(I.EQ.1) GO TO 1 | TIN00140 |
| IF(ALPHA.LT..5) TINORM=-TINORM | TIN00150 |
| RETURN | TIN00160 |
| 1 IFLAG=1 | TIN00170 |
| RETURN | TIN00180 |
| END | TIN00190 |

ORIGINAL PAGE IS
OF POOR QUALITY

13. DATA-TR PROCESSOR

FILE: DATATR

```

C      SUBROUTINE DATATR(ARRAY, TOP)                                DAT00010
C      IMPLICIT INTEGER(A-Z)                                       DAT00020
C                                                                    DAT00030
C      REAL    BIAS(16), BMAT(480), MAX(16), MIN(16), CON(16), CONMIN(32)  DAT00040
C                                                                    DAT00050
C      REAL    AMAX(16), AMIN(16), ACON(16)                        DAT00060
C                                                                    DAT00070
C                                                                    DAT00080
C      DIMENSION ARRAY(TOP), MAXPT(30), FILHIS(1616)              DAT00090
C                                                                    DAT00100
C      DIMENSION HDR1(15), HDR2(15), COMNT(15), INDATE(3)         DAT00110
C      INCLUDE COMRK1.LIST                                          DAT00120
C      INCLUDE COMRK9.LIST                                          DAT00130
C      INCLUDE COMRK6.LIST                                          DAT00140
C      COMMON/INFORM/NOCLS2, NOSUB2, NOFET2, VARSZ2, TOTVT2, NOFLD2,  DAT00150
C      AVAR2, COVAR2, CLSID2, SUBNO2, SUBDS2, FLOSV2, VERTX2,      DAT00160
C      FETVC2(30), SUBVC2(75), SURPTH(75), CLSVC2(60),           DAT00170
C      KEPPTS(60), NOGRP, GRPNAM(60), GRPDEX(61),                DAT00180
C      GRPCHK(61), GROUPS(124)                                     DAT00190
C      COMMON/GLOBAL/HEAD(63), MAPTAP, DATAPE, SAVTAP, BMFILE, BMKEY,  DAT00200
C      HISFIL, HISKEY, TRFORM, ERIPTP, ERPKEY, MAPUNT, NOFILE,    DAT00210
C      DRUMAD, ORMWOS, PAGSIZ, DATFIL, STAFIL, ASAV, ASAVFL,      DAT00220
C      ,NHSTUN, NHSTFI, SCTRUN, MAPFIL,                          DAT00230
C      ,DOTUNT, DOTFIL, NCHPAS, TRNSFL, BMTRFL, HISTFL, PCHUNT,   DAT00240
C      ,CRDUNT, PRTUNT, RANDIO                                     DAT00250
C DATA TRANSFORMATION COMMON BLOCK                                DAT00260
C COMMON/TRBLCK/OUTFMT, NOFEAT, FLDIRF(6), FETVEC(30)            DAT00270
CSEND DIMENSION VERTCS(2,11)                                       DAT00280
DATA BLANKS/' '/                                                  DAT00290
C                                                                    DAT00300
C                                                                    DAT00310
C                                                                    DAT00320
C                                                                    DAT00330
C                                                                    DAT00340
C                                                                    DAT00350
C                                                                    DAT00360
C                                                                    DAT00370
C                                                                    DAT00380
C                                                                    DAT00390
C                                                                    DAT00400
C                                                                    DAT00410
C                                                                    DAT00420
C                                                                    DAT00430
C                                                                    DAT00440
C                                                                    DAT00450
C                                                                    DAT00460
C                                                                    DAT00470
C                                                                    DAT00480
C                                                                    DAT00490
C                                                                    DAT00500
C                                                                    DAT00510
C                                                                    DAT00520
C                                                                    DAT00530
C                                                                    DAT00540
C                                                                    DAT00550
C                                                                    DAT00560
C                                                                    DAT00570
C                                                                    DAT00580
C                                                                    DAT00590
C                                                                    DAT00600
C                                                                    DAT00610
C                                                                    DAT00620
C                                                                    DAT00630
C                                                                    DAT00640
C                                                                    DAT00650
C                                                                    DAT00660
C                                                                    DAT00670
C                                                                    DAT00680
C                                                                    DAT00690
C                                                                    DAT00700
C                                                                    DAT00710
C                                                                    DAT00720
C                                                                    DAT00730
C                                                                    DAT00740
C                                                                    DAT00750
C                                                                    DAT00760
C                                                                    DAT00770
C                                                                    DAT00780
C                                                                    DAT00790

```

```

20 CALL KBTRAN ( BMAT, LCOMB, AQRAY, LAM, MAX, MIN, CON, TRANSF )
GO TO 50

IF RESCALING BY THE HISTOGRAM METHOD, OBTAIN THE PREDICTED
MAX AND MIN OF EACH COMPONENT OF THE TRANSFORMED DATA
(VIA MAXMAT ) AND PERFORM A HISTOGRAM OF THE TRANSFORMED
DATA ( VIA TRHIST ), IN ORDER TO OBTAIN THE RESCALING
PARAMETERS , CON AND MIN .

30 CALL MAXMAT ( AMAX, AMIN, ACON, BMAT, LCOMB, MAXPT )
CALL TRHIST ( ARRAY, AMAX, AMIN, ACON, BMAT, LCOMB, PEROUT,
* FILHIS, TOP, LAR, FLDNAM, NC, VERTCS, MAX, MIN, CON,
* BIAS )

APPLY TRANSFORMATION TO INPUT DATA, RESCALE ( IF OPTED ), APPLY
REJECTION ( PEROUT ) TO DISTRIBUTION OF TRANSFORMED DATA, AND
OUTPUT THE TRANSFORMED DATA ON THE FILE , TRFORM .

IF ( LAR.EQ.0 ) GO TO 60

50 CALL LNTRAN ( ARRAY, MAX, MIN, CON, BMAT, LCOMB, BMTRIG, SCAFLG, PEROUT,
* FILHIS, TOP, LAR, FLDNAM, NC, VERTCS, RESCAL, BIAS,
* NF , NPUN )

IF ( SCAFLG.EQ.1 ) GO TO 30

60 CONTINUE
DO 70 I=1,15
IPL3 = I * 3
HEAD(IPL3) = HDR1(I)
IPL29 = I * 29
HEAD(IPL29) = HDR2(I)
IPL47 = I * 47
70 HEAD(IPL47) = COMNT(I)

HEAD(22) = INDATE(1)
HEAD(23) = INDATE(2)
HEAD(24) = INDATE(3)

WRITE (6,80)
80 FORMAT(////////// 10X, '*** SDATA-TR COMPLETED ***' //)

RETURN
END

```

~~13-2~~
238

FILE: KBTRAN

```

      SUBROUTINE KBTRAN
      ( BMAT, LCOMB, ARRAY, LAM, MAX, MIN, EPS, TRANSF)
C
      IMPLICIT INTEGER(A-Z)
      REAL TMIN,TMAX
      REAL BMAT(480), MAX(16), MIN(16), EPS(16)
      REAL C(480), CC(480), D(16), DIAG(480), BMEAN(900)
C
      INCLUDE COMMK1.LIST
      INCLUDE COMMK9.LIST
      INCLUDE COMMK4.LIST
      INCLUDE COMMK6.LIST
      COMMON/INFORM/NOCLS2,NOSUR2,NOFET2,VARSZ2,TOTVT2,NOFLD2,
      * AVAR2,COVAR2,CLSID2,SURNO2,SURDS2,FLDSV2,VERTX2,
      * FETVC2(30),SUHVC2(75),SURPTR(75),CLSV2(60),
      * KEPPTS(60),NOGRP,GHPNAM(60),GRPDEX(61),
      * GRPCHK(61),GROUPS(124)
      DIMENSION HED1(15),HED2(15),DATE(3),COMENT(15)
      EQUIVALENCE (HED1(1),HEAD(4)),(DATE(1),HEAD(22)),
      * (HED2(1),HEAD(30)),(COMENT(1),HEAD(48))
      COMMON/GLOBAL/HEAD(63),MAPTAP,DATE,SAVTAP,BMFILE,BMKEY,
      * HISFIL,HISKEY,TRFORM,ERIPTR,ERPKEY,MAPUNT,NOFILE,
      * DRUMAD,DRUMDS,PAGSIZ,DATEFIL,STAFIL,ASAV,ASAVFL
      * ,NHSTUN,NHSTFI,SCRUN,MAPFIL
      * ,DOTUNT,DOTFIL,NCHPAS,TRANSFL,BMTRFL,HISTFL,PCHUNT,
      * CROUT,PRUNT,HANDIO
C DATA TRANSFORMATION COMMON BLOCK
      COMMON/TRRLCK/OUTFMT,NOFEAT,FLDINF(6), FETVEC(30)
C$END
C
      DIMENSION COVHD2(15)
      DIMENSION ARRAY(1)
      DIMENSION NSUB(75)
C
      DATA COVHD2/'...',',','TRAN','SFOK','MED ','STAT','ISTI','CS ','
      * '...',',',BLANKS/' '
C
      COMPUTE TRANSFORMED MEANS FOR EACH SURCLASS
C
      DO 10 I=1,15
      10 COVHD2(I) = BLANKS
      DO 20 I=1,NOSUR2
      IPP=SUHDS2+I-1
      20 NSUB(I)=ARRAY(IPP)
      LZ=AVAR2
      K=1
      DO 30 I=1,NOCLS2
      CALL MATVEC(BMAT,ARRAY(LZ),BMEAN(K),LCOMB,NOFET2)
      LZ=LZ+NOFET2
      30 K=K+LCOMB
C
      COMPUTE TRANSFORMED COVARIANCE MATRIX FOR EACH SUBCLASS
C
      KK=0
      K=COVAR2
      KKK=1
      DO 50 I=1,NOCLS2
      C MULTIPLY BMAT BY COVARIANCE MATRIX
      CALL MTMLSE(BMAT,ARRAY(K),C,LCOMB,NOFET2)
      C MULTIPLY RESULTING MATRIX BY TRANSPOSE OF BMAT
      CALL MTMDAT(C,BMAT,CC,LCOMB,NOFET2,LCOMB,(I),ARRAY(KKK))
      DO 40 II=1,LCOMB
      40 DIAG(KK+II)=D(II)
      KKK=KKK+(LCOMB*(LCOMB+1))/2
      KK=KK+LCOMB
      K=K+VARSZ2
      50 CONTINUE
C
      PRINT TRANSFORMED COVARIANCE MATRIX
C
      CV1=(LCOMB*(LCOMB+1))/2
C
      IF (TRANSF.EQ.0) GO TO 80
      DO 60 I=1,15
      TEMP = COMENT(I)
      COMENT(I) = COVHD2(I)
      60 COVHD2(I) = TEMP
C
      CALL PRTCOV(ARRAY(1),BMEAN(1),CV1,LCOMB,NSUB(1))

```

KBT00010
 KBT00020
 KBT00030
 KBT00040
 KBT00050
 KBT00060
 KBT00070
 KBT00080
 KBT00090
 KBT00100
 KBT00110
 KBT00120
 COM00010
 COM00020
 COM00030
 COM00040
 COM00050
 COM00010
 COM00020
 COM00030
 COM00040
 COM00050
 COM00010
 COM00020
 COM00030
 COM00040
 COM00050
 COM00060
 KBT00140
 KBT00150
 KBT00160
 KBT00170
 KBT00180
 KBT00190
 KBT00200
 KBT00210
 KBT00220
 KBT00230
 KBT00240
 KBT00250
 KBT00260
 KBT00270
 KBT00280
 KBT00290
 KBT00300
 KBT00310
 KBT00320
 KBT00330
 KBT00340
 KBT00350
 KBT00360
 KBT00370
 KBT00380
 KBT00390
 KBT00400
 KBT00410
 KBT00420
 KBT00430
 KBT00440
 KBT00450
 KBT00460
 KBT00470
 KBT00480
 KBT00490
 KBT00500
 KBT00510
 KBT00520
 KBT00530
 KBT00540
 KBT00550
 KBT00560
 KBT00570
 KBT00580
 KBT00590
 KBT00600
 KBT00610
 KBT00620
 KBT00630
 KBT00640

FILE: KSTRAN

```

      DO 70 I=1,15
      70 COMENT(I) = COVHD2(I)
      80 CONTINUE
C
C  CALCULATE MINIMUM AND MAXIMUM FOR EACH SUBCLASS
      DO 120 I=1,LCOMB
      DO 110 J=1,NOCLS2
        NEL = (J - 1) * LCOMB + I
        MAX(I) = RMEAN(NEL) * LAM * DIAG(NEL)
        IF (J.NE.1) GO TO 90
        TMAX = MAX(I)
      90 CONTINUE
        IF (MAX(I) .GE. TMAX) TMAX = MAX(I)
        MIN(I) = RMEAN(NEL) - LAM * DIAG(NEL)
        IF (J.NE.1) GO TO 100
        TMIN = MIN(I)
      100 CONTINUE
        IF (MIN(I) .LE. TMIN) TMIN = MIN(I)
      110 CONTINUE
        MIN(I) = TMIN
        MAX(I) = TMAX
        FPS(I) = 255. / (MAX(I) - MIN(I))
      120 CONTINUE
      RETURN
      END

```

KBT00650
 KBT00660
 KBT00670
 KBT00680
 KBT00690
 KBT00700
 KBT00710
 KBT00720
 KBT00730
 KBT00740
 KBT00750
 KBT00760
 KBT00770
 KBT00780
 KBT00790
 KBT00800
 KBT00810
 KBT00820
 KBT00830
 KBT00840
 KBT00850
 KBT00860
 KBT00870
 KBT00880
 KBT00890
 KBT00900

FILE LNTRAN

```

SUBROUTINE LNTRAN(IDATA,MAX,MIN,CON,BMAT,LCOMB,BMTRIG,SCAFLG,
* PEROUT,FILMIS,TOP,LAR,FLDNAM,NC,VERTCS, RESCAL, BIAS,
* NF, NPUN )
LNT00010
LNT00020
LNT00030
LNT00040
LNT00050
LNT00060
LNT00070
LNT00080
LNT00090
LNT00100
LNT00110
LNT00120
LNT00130
LNT00140
LNT00150
LNT00160
LNT00170
LNT00180
LNT00190
LNT00200
LNT00210
LNT00220
LNT00230
LNT00240
LNT00250
LNT00260
LNT00270
LNT00280
LNT00290
LNT00300
LNT00310
LNT00320
LNT00330
LNT00340
LNT00350
LNT00360
LNT00370
LNT00380
LNT00390
LNT00400
LNT00410
LNT00420
LNT00430
LNT00440
LNT00450
LNT00460
LNT00470
LNT00480
LNT00490
LNT00500
LNT00510
LNT00520
LNT00530
LNT00540
LNT00550
LNT00560
LNT00570
LNT00580
LNT00590
LNT00600
LNT00610
LNT00620
LNT00630
LNT00640
LNT00650
LNT00660
LNT00670
LNT00680
LNT00690
LNT00700
LNT00710
LNT00720
LNT00730
LNT00740
LNT00750
LNT00760

IF SCAFLG = 1 , RESCALE BY HISTOGRAM METHOD
IF SCAFLG = 2 , RESCALE BY THE STATISTICS METHOD
IF SCAFLG = 3 , RESCALE WITH USER-INPUT SCALING PARAMETERS

NOTE: IF THE FLAG, RESCAL , IS ZERO, NO RESCALING IS
PERFORMED. HOWEVER, PEROUT IS APPLIED TO THE TRANSFORMED
DATA DISTRIBUTION PRIOR TO FINAL OUTPUT OF TRANSFORMED
DATA VALUES .

IMPLICIT INTEGER(A-Z)
REAL TMIN(16), TMAX(16), MATOT , MITOT
REAL NEWMAX(16) , NEWMIN(16) , SUM , CUT
REAL NXCON,PMIN,PMAX,CMIN
REAL BIAS(16), XCON(16), XT(16), YREAL(16), NPER1, NPER2
REAL MAX(16), MIN(16), CON(16), BMAT(400), XXCON(16)
REAL MINSAV(16), MAXSAV(16), CONSAV(16)

DIMENSION TOTPTS(16), PMIN(16), PMAX(16)
DIMENSION HISBUF(101), VERTCS(2,11), FL(8)
DIMENSION IDATA(TOP), Y(8000), FILHIS(LCOMB,101)
DIMENSION RADMIN(16), RADMAX(16), MINCUT(16), MAXCUT(16)

DATA OP/'(//.CP//)'//.COMMA/'//
DATA TTL/'TOTL/'

INCLUDE COMMK1.LIST
INCLUDE COMMK9.LIST
INCLUDE COMMK4.LIST
INCLUDE COMMK6.LIST
COMMON/INFORM/NOCLS2,NOSUB2,NOFET2,VAWS2,TOTVT2,NOFLD2,
* AVAR2,COVAR2,CLS102,SUBNO2,SUMDS2,FLDSV2,VERTX2,
* FETVC2(30),SUBVC2(75),SUMPTR(75),CLSV2(60),
* KEPPTS(60),NOGRP,GRPNAM(60),GRPDEX(61),
* GRPCHK(61),GROUPS(124)
DIMENSION HED1(15),HED2(15),DATE(3),COMENT(15)
EQUIVALENCE (HED1(1),HEAD(4)),(DATE(1),HEAD(22)),
* (HED2(1),HEAD(30)),(COMENT(1),HEAD(48))
2 COMMON/GLOBAL/HEAD(63),MAPTAP,DATE,SAVTAP,HMFILE,BMKEY,
* HISFIL,HISKEY,TRFORM,ERIPY,EPKEY,MAPUNT,NOFILE,
* DPUMAD,DPMDIS,PAGSIZ,DATEFIL,SAFIL,ASAV,ASAVFL
* ,NHSTUN,NHSTFI,SCTRIN,MAPFIL
* ,DOTUNT,DATFIL,NCHPAS,TRNSFL,BMTRFL,HISTFL,PCHUNT,
* CROUNT,PWTUNT,WANDIO
C DATA TRANSFORMATION COMMON BLOCK
COMMON/TRBLCK/OUTFMT,NOFEAT,FLDINF(6), FETVEC(30)
C$END
C*** CODE ADDED JAN. 15,1979 TO ALLOW MULTI-FILE OUTPUT
C
RE=IND TRFORM
SKIP = NF - 1
CALL FSEFEL(TRFORM,SKIP,ISTAT)
IF (RESCAL.EQ.0) GO TO 50

CHECK FOR RESCALE FACTORS INPUT BY USER ( SCAFLG = 3 )
IF (SCAFLG.NE.3) GO TO 20

COMPUTE THE TRANSFORMED DATA MAX . USING INPUT
SCALING PARAMETERS , CON AND MIN .

DO 10 KF=1,LCOMB
MAX(KF) = 255./ CON(KF) * MIN(KF)
10 CONTINUE
20 CONTINUE

COMPUTE THE OUTPUT HISTOGRAM SCALE FACTOR. XCON

```

FILE LNTRAN

```

C
DO 30 KK=1,LCOMB
XCON(KK)=(MAX(KK)-MIN(KK))/80
30 FETVC2(KK)=KK
C
IF (SCAFLG.EQ.1) GO TO 80
C
FOR STATISTICAL OR INPUT SCALE PARAMETERS. SAVE THE INITIAL
SCALING PARAMETERS ( MIN. MAX. CON ) FOR RE-INITIALIZATION
OF THESE PARAMETERS ON THE SECOND AND SUCCEEDING FIELDS TO BE
INPUT , TRANSFORMED , AND RESCALED ( IF RESCAL GT 0 )
C
DO 40 I=1,LCOMB
MAXSAV(I) = MAX(I)
MINSAV(I) = MIN(I)
40 CONSAV(I) = CON(I)
C
POSITION THE INPUT DATA FILE, AND READ IN THE HEADER RECORD
50 CONTINUE
CALL TAPHDR(DATAP,DATAFIL)
C
GO TO 62
60 NF = NF + 1
C
READ THE COORDINATES ( VERTICES ) OF THE FIELD FOR THE DATA
TO BE TRANSFORMED .
C
62 LAM=LAREAD(FLDNAM,VERTCS,FLDINF,NC)
IF (LAM.EQ.0) GO TO 920
IF (LAM.LT.0) GO TO 900
C
FOR STATISTICAL OR INPUT SCALING PARAMETERS. INITIALIZE THE
SCALING PARAMETERS MAX , MIN , CON , XCON FOR THIS FIELD
C
IF (RESCAL.EQ.0) GO TO 80
C
DO 70 I=1,LCOMB
MAX(I) = MAXSAV(I)
MIN(I) = MINSAV(I)
CON(I) = CONSAV(I)
70 XCON(I) = ( MAX(I) - MIN(I) )/80.
80 CONTINUE
C
DO 90 I=1,LCOMB
MAXCUT(I) = 0
MINCUT(I) = 0
NEWMAX(I) = 255.0
XXCON(I) = XCON(I)
90 NEWMIN(I) = 0.0
C
MTHAN = 0
C
NSAMP=(FLDINF(5)-FLDINF(4))/FLDINF(6)+1
IDIM=NOFEAT*NSAMP
IF (IDIM.GT.TOP) GO TO 130
IN=NC-1
WRITE (6,100)
WRITE (6,110)FLDNAM,IN,FLDINF(6),FLDINF(3),((OP,VERTCS(1,K),COMMA,
*VERTCS(2,K),CP),K=1,IN)
100 FORMAT(14,124,'NO. OF SAMPLE LINE',/
*12,'FIELDNAME VERTICES INC INC VERTICES(SAMPLE,LINE)')
110 FORMAT(13X,44,4X,12,7X,14,2X,14,2X,5(A1,14,A1,14,A1,2X)/
* 5(A1,14,A1,14,A1,2X))
XDIM=LCOMB*NSAMP
C
IF (XDIM.LE.8000) GO TO 150
C
WRITE (6,120)

```

LNT00770
 LNT00780
 LNT00790
 LNT00800
 LNT00810
 LNT00820
 LNT00830
 LNT00840
 LNT00850
 LNT00860
 LNT00870
 LNT00880
 LNT00890
 LNT00900
 LNT00910
 LNT00920
 LNT00930
 LNT00940
 LNT00950
 LNT00960
 LNT00970
 LNT00980
 LNT00990
 LNT01000
 LNT01010
 LNT01020
 LNT01030
 LNT01040
 LNT01050
 LNT01060
 LNT01070
 LNT01080
 LNT01090
 LNT01100
 LNT01110
 LNT01120
 LNT01130
 LNT01140
 LNT01150
 LNT01160
 LNT01170
 LNT01180
 LNT01190
 LNT01200
 LNT01210
 LNT01220
 LNT01230
 LNT01240
 LNT01250
 LNT01260
 LNT01270
 LNT01280
 LNT01290
 LNT01300
 LNT01310
 LNT01320
 LNT01330
 LNT01340
 LNT01350
 LNT01360
 LNT01370
 LNT01380
 LNT01390
 LNT01400
 LNT01410
 LNT01420
 LNT01430
 LNT01440
 LNT01450
 LNT01460
 LNT01470
 LNT01480
 LNT01490
 LNT01500
 LNT01510
 LNT01520

FILE LNTRAN

```

120 FORMAT(//) THE NUMBER OF COMPONENTS TIMES THE NUMBER OF SAMPLES EXLNT01530
*CEEDS THE SIZE OF THE STORAGE AREA ***//) LNT01540
CALL CMERR LNT01550
130 WRITE (6,140) LNT01560
140 FORMAT(//) NUMBER OF CHANNELS TIMES NUMBER OF SAMPLES EXCEEDS 106 LNT01570
*00 ***//) LNT01580
CALL CMERR LNT01590
LNT01600
LNT01610
LNT01620
INITIALIZE TAPE READING FOR THIS FIELD LNT01630
150 CALL FLDINT(FLDINF,FETVEC,NOFEAT) LNT01640
LINES=(FLDINF(2)-FLDINF(1))/FLDINF(3)+1 LNT01650
LNT01660
LNT01670
WRITE HEADER RECORD ON OUTPUT FILE , TRFORM . LNT01680
LNT01690
LNT01700
CALL WRTHED(LCOMB,FETVC2,NSAMP,OUTFMT,TRFORM) LNT01710
LNT01720
LNT01730
DO 170 I=1,LCOMB LNT01740
BADMAX(I) = 0 LNT01750
BADMIN(I) = 0 LNT01760
PMIN(I) = 0 LNT01770
PMAX(I) = 0 LNT01780
TOTPTS(I) = 0 LNT01790
TMIN(I) = 1.0E35 LNT01800
TMAX(I) = -1.0E35 LNT01810
XCON(I) = XCON(I) LNT01820
DO 160 J=1,101 LNT01830
160 FILHIS(I,J) = 0 LNT01840
170 CONTINUE LNT01850
LNT01860
LNT01870
LNT01880
LSTLIN=0 LNT01890
M=0 LNT01900
180 M=M+1 LNT01910
IF (M.GT.LINES) GO TO 350 LNT01920
LNT01930
READ ONE SCAN LINE OF DATA FROM THE INPUT TAPE LNT01940
LNT01950
CALL LINEPD(IDATA,ENDTAP) LNT01960
IF (ENDTAP.NE.0) GO TO 350 LNT01970
IF (M.NE.1) GO TO 190 LNT01980
ILIN=FLDINF(1) LNT01990
GO TO 200 LNT02000
190 ILIN=ILIN+FLDINF(3) LNT02010
200 CONTINUE LNT02020
LNT02030
LNT02040
DETERMINE THE SAMPLE INTERCEPTS ON THE CURRENT SCAN LINE. LNT02050
WHICH ARE CONTAINED IN THE DESIRED FIELD BOUNDARIES. PLACE LNT02060
THE SAMPLE INTERCEPTS IN FL , AND THE NUMBER OF INTERCEPTS LNT02070
IN JJ . LNT02080
LNT02090
CALL FDLINT(VFRTCS,NC,FL,ILIN,NS,JJ) LNT02100
DO 210 K=1,NSAMP LNT02110
DO 210 IH=1,LCOMB LNT02120
ZSAMP=(IH-1)*NSAMP+K LNT02130
210 Y(ZSAMP)=0 LNT02140
LNT02150
LNT02160
LNT02170
LNT02180
LNT02190
LNT02200
NXCON = 255./100. LNT02210
LNT02220
LNT02230
LNT02240
LNT02250
LNT02260
LNT02270
LNT02280
LNT02290
LNT02300
LNT02310
LNT02320
LNT02330
LNT02340
LNT02350
LNT02360
LNT02370
LNT02380
LNT02390
LNT02400
LNT02410
LNT02420
LNT02430
LNT02440
LNT02450
LNT02460
LNT02470
LNT02480
LNT02490
LNT02500
LNT02510
LNT02520
LNT02530
LNT02540
LNT02550
LNT02560
LNT02570
LNT02580
LNT02590
LNT02600
LNT02610
LNT02620
LNT02630
LNT02640
LNT02650
LNT02660
LNT02670
LNT02680
LNT02690
LNT02700
LNT02710
LNT02720
LNT02730
LNT02740
LNT02750
LNT02760
LNT02770
LNT02780
LNT02790
LNT02800
LNT02810
LNT02820
LNT02830
LNT02840
LNT02850
LNT02860
LNT02870
LNT02880
LNT02890
LNT02900
LNT02910
LNT02920
LNT02930
LNT02940
LNT02950
LNT02960
LNT02970
LNT02980
LNT02990
LNT03000
LNT03010
LNT03020
LNT03030
LNT03040
LNT03050
LNT03060
LNT03070
LNT03080
LNT03090
LNT03100
LNT03110
LNT03120
LNT03130
LNT03140
LNT03150
LNT03160
LNT03170
LNT03180
LNT03190
LNT03200
LNT03210
LNT03220
LNT03230
LNT03240
LNT03250
LNT03260
LNT03270
LNT03280
LNT03290
LNT03300
LNT03310
LNT03320
LNT03330
LNT03340
LNT03350
LNT03360
LNT03370
LNT03380
LNT03390
LNT03400
LNT03410
LNT03420
LNT03430
LNT03440
LNT03450
LNT03460
LNT03470
LNT03480
LNT03490
LNT03500
LNT03510
LNT03520
LNT03530
LNT03540
LNT03550
LNT03560
LNT03570
LNT03580
LNT03590
LNT03600
LNT03610
LNT03620
LNT03630
LNT03640
LNT03650
LNT03660
LNT03670
LNT03680
LNT03690
LNT03700
LNT03710
LNT03720
LNT03730
LNT03740
LNT03750
LNT03760
LNT03770
LNT03780
LNT03790
LNT03800
LNT03810
LNT03820
LNT03830
LNT03840
LNT03850
LNT03860
LNT03870
LNT03880
LNT03890
LNT03900
LNT03910
LNT03920
LNT03930
LNT03940
LNT03950
LNT03960
LNT03970
LNT03980
LNT03990
LNT04000
LNT04010
LNT04020
LNT04030
LNT04040
LNT04050
LNT04060
LNT04070
LNT04080
LNT04090
LNT04100
LNT04110
LNT04120
LNT04130
LNT04140
LNT04150
LNT04160
LNT04170
LNT04180
LNT04190
LNT04200
LNT04210
LNT04220
LNT04230
LNT04240
LNT04250
LNT04260
LNT04270
LNT04280
LNT04290
LNT04300
LNT04310
LNT04320
LNT04330
LNT04340
LNT04350
LNT04360
LNT04370
LNT04380
LNT04390
LNT04400
LNT04410
LNT04420
LNT04430
LNT04440
LNT04450
LNT04460
LNT04470
LNT04480
LNT04490
LNT04500
LNT04510
LNT04520
LNT04530
LNT04540
LNT04550
LNT04560
LNT04570
LNT04580
LNT04590
LNT04600
LNT04610
LNT04620
LNT04630
LNT04640
LNT04650
LNT04660
LNT04670
LNT04680
LNT04690
LNT04700
LNT04710
LNT04720
LNT04730
LNT04740
LNT04750
LNT04760
LNT04770
LNT04780
LNT04790
LNT04800
LNT04810
LNT04820
LNT04830
LNT04840
LNT04850
LNT04860
LNT04870
LNT04880
LNT04890
LNT04900
LNT04910
LNT04920
LNT04930
LNT04940
LNT04950
LNT04960
LNT04970
LNT04980
LNT04990
LNT05000
LNT05010
LNT05020
LNT05030
LNT05040
LNT05050
LNT05060
LNT05070
LNT05080
LNT05090
LNT05100
LNT05110
LNT05120
LNT05130
LNT05140
LNT05150
LNT05160
LNT05170
LNT05180
LNT05190
LNT05200
LNT05210
LNT05220
LNT05230
LNT05240
LNT05250
LNT05260
LNT05270
LNT05280
LNT05290
LNT05300
LNT05310
LNT05320
LNT05330
LNT05340
LNT05350
LNT05360
LNT05370
LNT05380
LNT05390
LNT05400
LNT05410
LNT05420
LNT05430
LNT05440
LNT05450
LNT05460
LNT05470
LNT05480
LNT05490
LNT05500
LNT05510
LNT05520
LNT05530
LNT05540
LNT05550
LNT05560
LNT05570
LNT05580
LNT05590
LNT05600
LNT05610
LNT05620
LNT05630
LNT05640
LNT05650
LNT05660
LNT05670
LNT05680
LNT05690
LNT05700
LNT05710
LNT05720
LNT05730
LNT05740
LNT05750
LNT05760
LNT05770
LNT05780
LNT05790
LNT05800
LNT05810
LNT05820
LNT05830
LNT05840
LNT05850
LNT05860
LNT05870
LNT05880
LNT05890
LNT05900
LNT05910
LNT05920
LNT05930
LNT05940
LNT05950
LNT05960
LNT05970
LNT05980
LNT05990
LNT06000
LNT
```

FILE LNTRAN

```

C      CALL TRANSF TO DO A DATA TRANSFORMATION
C      CALL TRANSF
C      (XT, BMAT, IDATA, TOP, I, K, LCOMB, NSAMP, BIAS)
220 C CONTINUE
    DO 300 I=1,LCOMB
      IF ( XT(I) .LT. TMIN(I) ) TMIN(I) = XT(I)
      IF ( XT(I) .GT. TMAX(I) ) TMAX(I) = XT(I)

      IF RESCAL = 0 NO RESCALING IS APPLIED. OTHER WISE RESCALE
      USING SCALING PARAMETERS DERIVED FROM EITHER HISTOGRAM,
      STATISTICS, OR USER-INPUT ( SCAFLG= 1, 2, OR 3 )

      IF (RESCAL.GT.0) GO TO 260

      IF TRANSFORMED DATA IS NOT RESCALED,
      TEST FOR OUT- OF - RANGE TRANSFORMED VALUES

      SET = 0 ANY VALUE LESS THAN 0, OR LESS THAN THE NEW MIN
      AFTER APPLICATION OF PEROUT
      SET = 255 ANY VALUE GREATER THAN 255, OR GREATER THAN
      THE NEW MAX AFTER APPLICATION OF PEROUT

      IF (XT(I).LT.NEWMIN(I)) GO TO 230
      IF (XT(I).GT.NEWMAX(I)) GO TO 240
      GO TO 250
230 IF ( MTRAN .EQ. 0 ) BADMIN(I) = BADMIN(I) + 1
      XT(I) = 0.0
      GO TO 250
240 IF ( MTRAN .EQ. 0 ) BADMIN(I) = BADMIN(I) + 1
      XT(I) = 255.
250 CONTINUE

      FOR THE CURRENT SCAN LINE, HISTOGRAM THE TRANSFORMED DATA,
      AND STORE THE TRANSFORMED DATA INTO THE OUTPUT ARRAY, Y .

      DPT = XT(I)/XCON + 1.1
      TOTPTS(I) = TOTPTS(I) + 1
      IF ( DPT .GT. 101 ) DPT = 101
      IF ( DPT .LE. 0 ) DPT = 1
      FILHIS(I,DPT) = FILHIS(I,DPT) + 1
      ZS = (I-1) * NSAMP * K
      Y(ZS) = XT(I) * 0.5
      GO TO 300
260 CONTINUE

      FOR THE CURRENT SCAN LINE, HISTOGRAM THE TRANSFORMED DATA, AND
      STORE THE TRANSFORMED DATA INTO THE OUTPUT ARRAY, Y .

      IF (XT(I).LT.MIN(I)) GO TO 270
      IF (XT(I).GT.MAX(I)) GO TO 280
      YREAL(I)=CON(I)*(XT(I)-MIN(I))

      DPT = ( XT(I) - MIN(I) ) / XCON(I) + 11
      IF ( DPT .LE. 0 ) DPT = 1
      IF ( DPT .GT. 101 ) DPT = 101

      GO TO 290
270 DPT = ABS( MIN(I) - XT(I) ) / XCON(I)
      DPT = 10 - DPT
      PMIN(I) = PMIN(I) + 1
      IF (DPT.LE.0) DPT=1
      YREAL(I)=0
      GO TO 290
280 DPT = ABS( XT(I) - MAX(I) ) / XCON(I)
      DPT = DPT + 91

```

LNT02290
 LNT02300
 LNT02310
 LNT02320
 LNT02330
 LNT02340
 LNT02350
 LNT02360
 LNT02370
 LNT02380
 LNT02390
 LNT02400
 LNT02410
 LNT02420
 LNT02430
 LNT02440
 LNT02450
 LNT02460
 LNT02470
 LNT02480
 LNT02490
 LNT02500
 LNT02510
 LNT02520
 LNT02530
 LNT02540
 LNT02550
 LNT02560
 LNT02570
 LNT02580
 LNT02590
 LNT02600
 LNT02610
 LNT02620
 LNT02630
 LNT02640
 LNT02650
 LNT02660
 LNT02670
 LNT02680
 LNT02690
 LNT02700
 LNT02710
 LNT02720
 LNT02730
 LNT02740
 LNT02750
 LNT02760
 LNT02770
 LNT02780
 LNT02790
 LNT02800
 LNT02810
 LNT02820
 LNT02830
 LNT02840
 LNT02850
 LNT02860
 LNT02870
 LNT02880
 LNT02890
 LNT02900
 LNT02910
 LNT02920
 LNT02930
 LNT02940
 LNT02950
 LNT02960
 LNT02970
 LNT02980
 LNT02990
 LNT03000
 LNT03010
 LNT03020
 LNT03030
 LNT03040

FILE LNTRAN

```

      PMAX(I) = PMAX(I) * 1
      IF (OPT.GT.101) OPT=101
      YHEAL(I)=255
290  TOTPTS(I)=TOTPTS(I)+1
      FILHIS(I,OPT)=FILHIS(I,OPT)+1
      ZSAMP=(I-1)*NSAMP+K
      Y(ZSAMP) = YHEAL(I) * 0.5
300  CONTINUE
      GO TO 330
310  IF (JKP1.GE.JJ) GO TO 340
320  CONTINUE
330  CONTINUE
340  CONTINUE
      IF (M.EU.LINES) LSTLIN=-1

      OUTPUT ONE LINE OF TRANSFORMED DATA ON THE OUTPUT FILE , TRFORM

      CALL WPTLN(Y,LSTLIN)
      GO TO 180

      IF RESCALING THE TRANSFORMED DATA BY EITHER THE STATISTICAL
      OR USER-INPUT SCALING PARAMETERS.
      APPLY PEROUT ( OF POINTS TO BE REJECTED) TO THE TRANSFORMED
      DATA DISTRIBUTION - OBTAIN THE MAX. AND SCALING PARAMETERS
      MIN AND CON. AFTER APPLICATION OF PEROUT ( ALSO, THE HISTOGRAM
      SCALE FACTOR , XCON ) .

      IF NOT RESCALING, APPLY PEROUT TO THE TRANSFORMED DATA.
      GET NEW MAX AND MIN. RE-HISTOGRAM , AND OUTPUT THE REVISED DISTR.

350  CONTINUE
      IF (PEROUT.LE.0) GO TO 600
      IF (SCAFLG.EQ.1) GO TO 600
      IF (MTWAN.EQ.1) GO TO 600
      IF (RESCAL.GT.0) GO TO 430
      NPER1 = FLOAT(PEROUT)/200.0
      DO 420 I=1,LCOMB
      CUT = NPER1 * FLOAT( TOTPTS(I) )
      SUM = 0.0
      DO 370 J=1,101.1
      IF (SUM.GE.CUT) GO TO 360
      GO TO 370
360  MINCUT(I) = SUM
      NEWMIN(I) = (J-1) * NXCON * 0.5
      GO TO 380
370  SUM = SUM + FILHIS(I,J)
380  SUM = 0.0
      J=101
385  J=J-1
      IF (SUM.GE.CUT) GO TO 390
      GO TO 400
390  MAXCUT(I) = SUM
      NEWMAX(I) = ( J - 1 ) * NXCON * 0.5
      GO TO 410
400  SUM = SUM + FILHIS(I,J)
      IF (J.GT.1) GO TO 385
410  CONTINUE
420  CONTINUE
      GO TO 580

C
430  NPER1 = PEROUT * .01 * .001
      NPER2 = PEROUT * .01 * .001

C
      RSET=0
      IG = 0
      IH = 0

```

ORIGINAL PAGE IS
OF POOR QUALITY

LNT03050
 LNT03060
 LNT03070
 LNT03080
 LNT03090
 LNT03100
 LNT03110
 LNT03120
 LNT03130
 LNT03140
 LNT03150
 LNT03160
 LNT03170
 LNT03180
 LNT03190
 LNT03200
 LNT03210
 LNT03220
 LNT03230
 LNT03240
 LNT03250
 LNT03260
 LNT03270
 LNT03280
 LNT03290
 LNT03300
 LNT03310
 LNT03320
 LNT03330
 LNT03340
 LNT03350
 LNT03360
 LNT03370
 LNT03380
 LNT03390
 LNT03400
 LNT03410
 LNT03420
 LNT03430
 LNT03440
 LNT03450
 LNT03460
 LNT03470
 LNT03480
 LNT03490
 LNT03500
 LNT03510
 LNT03520
 LNT03530
 LNT03540
 LNT03550
 LNT03560
 LNT03570
 LNT03580
 LNT03590
 LNT03600
 LNT03610
 LNT03620
 LNT03630
 LNT03640
 LNT03650
 LNT03660
 LNT03670
 LNT03680
 LNT03690
 LNT03700
 LNT03710
 LNT03720
 LNT03730
 LNT03740
 LNT03750
 LNT03760
 LNT03770
 LNT03780
 LNT03790
 LNT03800

FILE LNTRAN

```

      IE = 0
      IA = 0
C
      DO 560 I=1,LCOMB
      MATOT=TOTPTS(I)*NPER1
      MITOT=TOTPTS(I)*NPER2
      IF (PMIN(I).GT.MATOT) GO TO 440
      GO TO 460
C RESET MIN SMALLER
      440 CMIN=PMIN(I)
      J=10
      445 J=J-1
      IG=IG+1
      CMIN=CMIN-FILHIS(I,J)
      IF (CMIN.GT.MATOT) GO TO 450
      IF (CMIN.LT.MITOT) IG=IG-1
      RSET=1
      MIN(I)=MIN(I)-IG*XCON(I)
      IG=0
      GO TO 490
      450 IF (J.GT.1) GO TO 445
      460 IF (PMIN(I).LT.MITOT) GO TO 470
      GO TO 490
C RESET MIN LARGER
      470 CMIN=PMIN(I)
      DO 480 J=11,91
      IH=IH+1
      CMIN=CMIN+FILHIS(I,J)
      IF (CMIN.LT.MITOT) GO TO 480
      IF (CMIN.GT.MATOT) IH=IH-1
      MIN(I)=MIN(I)+IH*XCON(I)
      IH=0
      RSET=1
      GO TO 490
      480 CONTINUE
C CHECK MAX
      490 CONTINUE
      IF (PMAX(I).GT.MATOT) GO TO 500
      GO TO 520
C RESET MAX LARGER
      500 CMIN=PMAX(I)
      DO 510 J=92,101
      IE=IE+1
      CMIN=CMIN-FILHIS(I,J)
      IF (CMIN.GT.MATOT) GO TO 510
      IF (CMIN.LT.MITOT) IE=IE-1
      RSET=1
      MAX(I)=MAX(I)+IE*XCON(I)
      IE=0
      GO TO 550
      510 CONTINUE
C RESET MAX SMALLER
      520 IF (PMAX(I).LT.MITOT) GO TO 530
      GO TO 550
      530 CMIN=PMAX(I)
      J=91
      535 J=J-1
      IA=IA+1
      CMIN=CMIN+FILHIS(I,J)
      IF (CMIN.LT.MITOT) GO TO 540
      IF (CMIN.GT.MATOT) IA=IA-1
      RSET=1
      MAX(I)=MAX(I)-IA*XCON(I)
      IA=0
      GO TO 550
      540 IF (J.GT.1) GO TO 535
      550 CONTINUE
      560 CONTINUE
      IF (RSET.EQ.0) GO TO 600
      DO 570 I=1,LCOMB
      XCON(I)=(MAX(I)-MIN(I))/40
      XXCON(I) = XCON(I)
      570 CON(I) = 255. / (MAX(I) - MIN(I))
C
      580 IF (NF.EQ.1) GO TO 590
C

```

LNT03810
 LNT03820
 LNT03830
 LNT03840
 LNT03850
 LNT03860
 LNT03870
 LNT03880
 LNT03890
 LNT03900
 LNT03910
 LNT03920
 LNT03930
 LNT03940
 LNT03950
 LNT03960
 LNT03970
 LNT03980
 LNT03990
 LNT04000
 LNT04010
 LNT04020
 LNT04030
 LNT04040
 LNT04050
 LNT04060
 LNT04070
 LNT04080
 LNT04090
 LNT04100
 LNT04110
 LNT04120
 LNT04130
 LNT04140
 LNT04150
 LNT04160
 LNT04170
 LNT04180
 LNT04190
 LNT04200
 LNT04210
 LNT04220
 LNT04230
 LNT04240
 LNT04250
 LNT04260
 LNT04270
 LNT04280
 LNT04290
 LNT04300
 LNT04310
 LNT04320
 LNT04330
 LNT04340
 LNT04350
 LNT04360
 LNT04370
 LNT04380
 LNT04390
 LNT04400
 LNT04410
 LNT04420
 LNT04430
 LNT04440
 LNT04450
 LNT04460
 LNT04470
 LNT04480
 LNT04490
 LNT04500
 LNT04510
 LNT04520
 LNT04530
 LNT04540
 LNT04550
 LNT04560

FILE LNTRAN

ORIGINAL PAGE IS
OF POOR QUALITY

```

      REWIND TRFORM
      SKIP=NF-1
      CALL F5FMFL (TRFORM,SKIP,ISTAT)
      MTRAN=1
      GO TO 150
590  REWIND TRFORM
      MTRAN=1
      GO TO 150
600  CONTINUE
      MTRAN=0
C
      IF (RESCAL.EQ.0) GO TO 710
C
      WRITE (6,610)NF
610  FORMAT (////// 22X, '* OUTPUT FILE ', I3, 1X, '*' // 5X,
      * '*** TRANSFORMED VALUES RESCALED TO A RANGE 0 - 255 ***' )
C
      IF (SCAFLG.EQ.1) *PITE (6,620)
      IF (SCAFLG.EQ.2) WRITE (6,630)
      IF (SCAFLG.EQ.3) WRITE (6,640)
C
620  FORMAT ( 24X, '(HISTOGRAM METHOD)' / )
630  FORMAT ( 23X, '(STATISTICS METHOD)' / )
640  FORMAT ( 20X, '(INPUT SCALING PARAMETERS)' / )
C
      WRITE (6,650)
C
650  FORMAT (////// 7X, '... ORIGINAL TRANSFORMED DATA RANGE ...' //
      * 'T11, 'MIN', T32, 'MAX', T51, '(BIAS)' // )
C
      DO 660 M=1,LCOMB
660  WRITE (6,670)TMIN(M),TMAX(M),BIAS(M)
C
670  FORMAT ( 5X, F11.4, 10X, F11.4, 9X, '( ', F11.4, 1X, ') ' / )
C
      WRITE (6,680)
680  FORMAT (////// 7X, '... TRANSFORMED DATA RANGE, AFTER APPLICATION OF
      * PEROUT = 255/(MAX-MIN) ' // )
      * 'CON = 255/(MAX-MIN) ' // )
C
      PRINT OUT NEW MAX,MIN,CON ARRAYS
      DO 690 M=1,LCOMB
690  WRITE (6,700)MIN(M),MAX(M),CON(M)
700  FORMAT (5X,3(F8.4,5X))
C
      GO TO 850
C
710  IF (PEROUT.GT.0) GO TO 730
C
      DO 720 I=1,LCOMB
      IF (HADMIN(I).EQ.0) NEWMIN(I) = TMIN(I)
      IF (HADMAX(I).EQ.0) NEWMAX(I) = TMAX(I)
720  CONTINUE
C
730  WRITE (6,740)NF
740  FORMAT (////// 19X, '* OUTPUT FILE ', I3, 1X, '*' )
C
      WRITE (6,750)
750  FORMAT (////// 5X, '*** TRANSFORMED VALUES NOT RESCALED ***' // )
C
      WRITE (6,760)LCOMB,(TMIN(I),I=1,LCOMB)
760  FORMAT (// 5X, 'TRANSFORMED MINIMUMS, COMPONENTS 1-' , I2, 2X,
      * '....' // 2(5X, HF12.2 / ) )
C
      WRITE (6,770)LCOMB,(TMAX(I),I=1,LCOMB)
770  FORMAT (// 5X, 'TRANSFORMED MAXIMUMS, COMPONENTS 1-' , I2, 2X,
      * '....' // 2( 5X, HF12.2 / ) )
C
      WRITE (6,780)LCOMB,(BIAS(I),I=1,LCOMB)
780  FORMAT (/// 5X, 'TRANSFORMED VALUE BIAS, COMPONENTS 1 - ' , I4, 2X,
      * '...' // 2( 5X, HF12.2 / ) )
C

```

LNT04570
 LNT04580
 LNT04590
 LNT04600
 LNT04610
 LNT04620
 LNT04630
 LNT04640
 LNT04650
 LNT04660
 LNT04670
 LNT04680
 LNT04690
 LNT04700
 LNT04710
 LNT04720
 LNT04730
 LNT04740
 LNT04750
 LNT04760
 LNT04770
 LNT04780
 LNT04790
 LNT04800
 LNT04810
 LNT04820
 LNT04830
 LNT04840
 LNT04850
 LNT04860
 LNT04870
 LNT04880
 LNT04890
 LNT04900
 LNT04910
 LNT04920
 LNT04930
 LNT04940
 LNT04950
 LNT04960
 LNT04970
 LNT04980
 LNT04990
 LNT05000
 LNT05010
 LNT05020
 LNT05030
 LNT05040
 LNT05050
 LNT05060
 LNT05070
 LNT05080
 LNT05090
 LNT05100
 LNT05110
 LNT05120
 LNT05130
 LNT05140
 LNT05150
 LNT05160
 LNT05170
 LNT05180
 LNT05190
 LNT05200
 LNT05210
 LNT05220
 LNT05230
 LNT05240
 LNT05250
 LNT05260
 LNT05270
 LNT05280
 LNT05290
 LNT05300
 LNT05310
 LNT05320

FILE LNTRAN

```

C
C
C      WRITE (6,790) (I,HADMIN(I),I=1,LCOMB)
C      790 FORMAT(/// 5X,'NO. OF TRANSFORMED VALUES LESS THAN 0 ( SET = 0 ) :
C      * // (5X,'COMPONENT', 1X, I2, '...', 16, 2X, 'VALUES' ) )
C
C      WRITE (6,800) (I,HADMAX(I),I=1,LCOMB)
C      800 FORMAT(/// 5X, 'NO. OF TRANSFORMED VALUES GREATER THAN 255 ( SET =
C      * 255 ) : ' // (5X,'COMPONENT', 1X, I2, '...', 16, 2X, 'VALUES' ) )
C
C      NPER1 = FLOAT(PEROUT)/2.0
C      WRITE (6,810) NPER1, LCOMB, (MINCUT(I), I=1, LCOMB)
C      810 FORMAT(/// 3X, 'NO. OF LOWER TAIL POINTS REJECTED ( SET = 0 FL
C      * OR OUTPUT ) TO SATISFY', F6.1, 2X, ' - CUT-OFF, COMPONENTS 1 - '
C      * 14, 2X, '...' // 16(I8) )
C
C      WRITE (6,820) NPER1, LCOMB, (MAXCUT(I), I=1, LCOMB)
C      820 FORMAT(/// 3X, 'NO. OF UPPER TAIL POINTS REJECTED ( SET = 255 FL
C      * OR OUTPUT ) TO SATISFY', F6.1, 2X, ' - CUT-OFF, COMPONENTS 1 - '
C      * 14, 2X, '...' // 16(I8) )
C      NPCT = 100 - PEROUT
C      WRITE (6,830) NPCT, LCOMB, (NEWMIN(I), I=1, LCOMB)
C      830 FORMAT(/// 3X, '**** FINAL OUTPUT TRANSFORMED VALUES, CENTRAL',
C      * 15, 3X, ' ', 2X, 'OF DISTRIBUTION : ' // 3X, 'MINIMUMS, COMPONENTS
C      * 1 - ' // 14, 2X, '...' // 2(5X, 8F12.2 / ) )
C
C      WRITE (6,840) LCOMB, (NEWMAX(I), I=1, LCOMB)
C      840 FORMAT(/// 3X, 'MAXIMUMS, COMPONENTS 1 - ' // 14, 2X, '...' //
C      * 2( 5X, 8F12.2 / ) )
C
C      PRINT HISTOGRAMS
C      850 CONTINUE
C
C      XSIZ=101
C      XHGH=255
C      XLOW=0
C      YSIZ=15
C      CALL COMHST(FILHIS,HISBUF,TTL,LCOMB,FETVC2,XSIZ,XHGH,XLOW,YSIZ)
C
C      IF (RESCAL.EQ.0) GO TO 60
C
C      WRITE (6,860) NF
C      860 FORMAT(1H1 // 5X,'SCALING PARAMETERS USED ON TRANSFORMED VALUES, O
C      * OUTPUT FILE', 15// 19X, 'MINIMUM', 7X, 'MAXIMUM', 7X,
C      * 'SCALE FACTOR ( CON )' )
C
C      WRITE (6,870) (FETVC2(IL), MIN(IL), MAX(IL), CON(IL), IL=1, LCOMB)
C      870 FORMAT(1X, 'COMPONENT', 13, 1X, F12.3, 2X, F12.3, 7X, F12.3 )
C
C      IF (NPUN.LE.0) GO TO 890
C
C      PUNCH 880, (CON(MN), MIN(MN), MN=1, LCOMB)
C      880 FORMAT( ('OPTION', 4X, 'SCAFAC=', 2( ' (', F9.3, ' ', F9.3,
C      * ' ) , ' ) ) )
C      890 CONTINUE

```

LNT05330
 LNT05340
 LNT05350
 LNT05360
 LNT05370
 LNT05380
 LNT05390
 LNT05400
 LNT05410
 LNT05420
 LNT05430
 LNT05440
 LNT05450
 LNT05460
 LNT05470
 LNT05480
 LNT05490
 LNT05500
 LNT05510
 LNT05520
 LNT05530
 LNT05540
 LNT05550
 LNT05560
 LNT05570
 LNT05580
 LNT05590
 LNT05600
 LNT05610
 LNT05620
 LNT05630
 LNT05640
 LNT05650
 LNT05660
 LNT05670
 LNT05680
 LNT05690
 LNT05700
 LNT05710
 LNT05720
 LNT05730
 LNT05740
 LNT05750
 LNT05760
 LNT05770
 LNT05780
 LNT05790
 LNT05800
 LNT05810
 LNT05820
 LNT05830
 LNT05840
 LNT05850
 LNT05860
 LNT05870
 LNT05880
 LNT05890
 LNT05900
 LNT05910
 LNT05920
 LNT05930
 LNT05940
 LNT05950
 LNT05960
 LNT05970
 LNT05980
 LNT05990
 LNT06000
 LNT06010
 LNT06020
 LNT06030
 LNT06040
 LNT06050
 LNT06060
 LNT06070
 LNT06080

FILE LNTRAN

| | | | |
|-----|--|--------|----------|
| C | IF (SCAFLG .EQ. 1) | RETURN | LNT06090 |
| C | | | LNT06100 |
| C | | | LNT06110 |
| | GO TO 60 | | LNT06120 |
| 900 | IF (LAM.EQ.0) GO TO 920 | | LNT06130 |
| | WRITE (6,910)FLDNAM | | LNT06140 |
| 910 | FORMAT(//// 5X, '***** DATATR/LNTRAN ***** ERROR ON INPUT FIELD DE | | LNT06150 |
| | *FINITION CARD. FOR FIELD NAME ' ,1H', A4, 1H', 3X, '*****' / | | LNT06160 |
| | * 10X, 'CONTINUING TO NEXT FIELD DEFINITION CARD(S): //// ') | | LNT06170 |
| | GO TO 60 | | LNT06180 |
| 920 | CONTINUE | | LNT06190 |
| | RETURN | | LNT06200 |
| | END | | LNT06210 |
| | | | LNT06220 |

ORIGINAL PAGE 1
OF POOR QUALITY

FILE: MAXMAT

| | | |
|---|---|----------|
| C | SUBROUTINE MAXMAT (MAX, MIN, CON, BMAT, LCOMB, MAXPT) | MXM00010 |
| C | COMPUTE AN APPROXIMATE TRANSFORMED MAX AND MIN FOR EACH COMPONENT | MXM00020 |
| C | OF THE TRANSFORMATION | MXM00030 |
| C | IMPLICIT INTEGER(A-Z) | MXM00040 |
| C | DIMENSION MAXPT(30) | MXM00050 |
| C | | MXM00060 |
| C | REAL BMAT(480), MAX(16), MIN(16), CON(16) | MXM00070 |
| C | | MXM00080 |
| C | | MXM00090 |
| C | | MXM00100 |
| C | USING INPUT (OR DEFAULT) MAXIMUM DATA VALUE FOR EACH | MXM00110 |
| C | CHANNEL, COMPUTE THE TRANSFORMED VALUE RANGE (MAX AND MIN) | MXM00120 |
| C | AND COMPUTE THE HISTOGRAM SCALING FACTOR, CON . | MXM00130 |
| C | | MXM00140 |
| C | | MXM00150 |
| C | | MXM00160 |
| C | INCLUDE COMRK9.LIST | MXM00170 |
| C | DATA TRANSFORMATION COMMON BLOCK | MXM00180 |
| C | COMMON/TRHLCK/OUTFMT,NOFEAT,FLDINF(6), FETVEC(30) | |
| C | CSEND | MXM00200 |
| C | DO 30 I=1,LCOMB | MXM00210 |
| C | MAX(I) = 0.0 | MXM00220 |
| C | MIN(I) = 0.0 | MXM00230 |
| C | DO 20 J=1,NOFEAT | MXM00240 |
| C | K=(J-1)*LCOMB+I | MXM00250 |
| C | IF (BMAT(K).LE.0.0) GO TO 10 | MXM00260 |
| C | MAX(I) = MAX(I) + BMAT(K) * MAXPT(J) | MXM00270 |
| C | | MXM00280 |
| C | GO TO 20 | MXM00290 |
| C | 10 CONTINUE | MXM00300 |
| C | | MXM00310 |
| C | MIN(I) = MIN(I) + BMAT(K) * MAXPT(J) | MXM00320 |
| C | | MXM00330 |
| C | 20 CONTINUE | MXM00340 |
| C | CON(I)=(MAX(I)-MIN(I))/100. . | MXM00350 |
| C | 30 CONTINUE | MXM00360 |
| C | RETURN | MXM00370 |
| C | END | MXM00380 |
| | | MXM00390 |

FILE: SETREM

| | | |
|----|--|----------|
| | SUPROUTINE SETREM | SRE00010 |
| | (CONMIN, CON, MIN, ADDNUM, LCOMB) | SRE00020 |
| C | IMPLICIT INTEGER (A-Z) | SRE00030 |
| | REAL CON(16), MIN(16), CONMIN(2, 16) | SRE00040 |
| | NUMCM = ADDNUM / 2 | SRE00050 |
| | IF (NUMCM.NE.LCOMB) GO TO 20 | SRE00060 |
| | DO 10 NM=1,LCOMB | SRE00070 |
| | CON(NM) = CONMIN(1, NM) | SRE00080 |
| | MIN(NM) = CONMIN(2, NM) | SRE00090 |
| 10 | CONTINUE | SRE00100 |
| | RETURN | SRE00110 |
| 20 | WRITE (6,30) NUMCM, LCOMB | SRE00120 |
| 30 | FORMAT (5X, 'SETREM ERROR - THERE WERE ', I5, ' SCALE ', | SRE00130 |
| C | 'FACTORS AND MINIMUM VALUES INPUT THROUGH THE SCAFAC ', | SRE00140 |
| C | 'OPTION: ', | SRE00150 |
| C | /, 5X, I5, ' LINEAR COMBINATIONS WERE REQUESTED. ', | SRE00160 |
| C | /, 5X, ' THERE MUST BE A SCALE FACTOR AND A MINIMUM ', | SRE00170 |
| C | 'VALUE FOR EACH LINEAR COMBINATION. ', | SRE00180 |
| C | /, 5X, ' THE PROGRAM WILL TERMINATE THROUGH CMERR') | SRE00190 |
| | CALL CMERR | SRE00200 |
| | END | SRE00210 |
| | | SRE00220 |

ORIGINAL PAGE IS
OF POOR QUALITY

FILE: SETUP8

```

SUBROUTINE SETUP8(BMAT,LCOMB,BMTRIG,PEROUT,MAXPT,ARRAY,LAM,SCAFLG,SET00010
* TOP, TRANSF, RESCAL, BIAS, ADDNUM, CONMIN,NPUN,NF) SET00020
C IMPLICIT INTEGER(A-Z) SET00030
C LOGICAL NUDTAP, NUDFIL, NUSTAP, NUSFIL SET00040
C REAL CONMIN(32), BIAS(16), BMAT(480) SET00050
C DIMENSION MAXPT(30) SET00060
C DIMENSION ARRAY(1) SET00070
C DIMENSION EQUVEC(2) SET00080
C DIMENSION CINDEX(19), SET00090
C SINVEC(3), FRVEC1(3), FRVEC2(3), CARD2(62), SET00100
C BTEST(3) SET00110
C DIMENSION COVHD1(15), OP(2), CP(2) SET00120
C DIMENSION MTX(5),ACARD(20) SET00130
C INCLUDE COMAK1.LIST SET00140
C INCLUDE COMAK4.LIST SET00150
C INCLUDE COMAK6.LIST SET00160
C INCLUDE COMAK9.LIST SET00170
COMMON/INFORM/NOCLS2,NOSUR2,NOFET2,VARSZ2,TOTVT2,NOFLD2, SET00180
* AVAR2,COVAR2,CLSD2,SURN02,SUBDS2,FLDSV2,VENTX2, SET00190
* FETVC2(30),SUBVC2(75),SUBPTR(75),CLSV2(60), SET00200
* KEPPTS(60),NOGRP,GRPNAM(60),GRPDEX(61), SET00210
* GRPCHK(61),GROUPS(124) SET00220
C DIMENSION HED1(15),HED2(15),DATE(3),COMENT(15) SET00230
C EQUIVALENCE (HED1(1),HEAD(4)),(DATE(1),HEAD(22)), SET00240
C (HED2(1),HEAD(30)),(COMENT(1),HEAD(48)), SET00250
C COMMON/GLOBAL/HEAD(63),MAPTAP,DATAPE,SAVTAP,BMFILE,BMKEY, SET00260
C HISFIL,HISKEY,TRFORM,ERPTP,ERPKEY,MAPUNT,NOFILE, SET00270
C DRUMAD,DRMWD5,PAGSIZ,DATFIL,STAFIL,ASAV,ASAVFL SET00280
C ,NHSTUN,NHSTFI,SCTRUN,MAPFIL SET00290
C ,DOTUNT,DOTFIL,NCHPAS,TRNSFL,BMTRFL,HISTFL,PCHUNT, SET00300
C CRDUNT,PKTUNT,RANDIO SET00310
C DATA TRANSFORMATION COMMON BLOCK SET00320
COMMON/TRBLCK/OUTFMT,NOFEAT,FLDINF(6), FETVEC(30) SET00330
C SEND SET00340
C EQUIVALENCE (FLDINF(1),LINSTR),(FLDINF(2),LINEND), SET00350
C (FLDINF(3),LININC),(FLDINF(4),SAMSTR), SET00360
C (FLDINF(5),SAMEND),(FLDINF(6),SAMINC) SET00370
C DATA CINDEX/'R-MA','CHAN','FORM','HED1','HED2', SET00380
C ,DATE','COMM','MAXP','PERO','SUBC', SET00390
C ,MODU','LAM','OPTI','END','DATA','STAT', SET00400
C ,RESC','BIAS','TROU' SET00410
C DATA MTX / 4, '0', '1', 'S', 'P' / SET00420
C DATA OP / 1, '1' / , CP / 1, '1' / , ZERO / 0 / SET00430
C DATA EQUVEC / 1, '1' / SET00440
C DATA SINVEC / 2, '1', '1' / SET00450
C CINMAX / 19, MAXFET / 30, SET00460
C FRVEC1 / 2, '1', '0' / , FRVEC2 / 2, 'U', 'L' / , BLANK / ' ' / SET00470
C BTEST / 2, 'C', 'F' / SET00480
C CBCD / 'C' / , FRCD / 'F' / , UBOD / 'U' / SET00490
C DATA COVHD1 / '... ' , 'ORIG', 'INAL', 'STA', 'TIST', 'ICS ' , '... ' / SET00500
C INITIALIZE FLAGS AND DEFAULT VALUES SET00510
C NOSUR2=0 SET00520
C NOGRP=0 SET00530
C BMTRIG=0 SET00540
C NSF = 1 SET00550
C RESCAL = 0 SET00560
C SCAFLG = 0 SET00570
C MPT = 0 SET00580
C ORIG = 0 SET00590
C TRANSF = 0 SET00600
C NPUN = 0 SET00610
C OUTFMT=2 SET00620
C TRFORM=14 SET00630
C INITIALIZE THE TRANSFORMATION BIAS VECTOR ( BIAS ) AND NO. OF SET00640
C BIAS VALUES ( NBS ) SET00650
C DO 10 I=1,16 SET00660
C 10 BIAS(I) = 0.0 SET00670

```

ORIGINAL PAGE IS
OF POOR QUALITY

FILE: SETUP8

```

      NBS = 0
      NUDTAP = .FALSE.
      NUDFIL = .FALSE.
      NUSTAP = .FALSE.
      NUSFIL = .FALSE.
      BMSWT=1

C      INITIALIZE THE MAXIMUM EXPECTED DATA VALUE, FOR EACH CHANNEL
C
      DO 20 I=1,30
      20 MAXPT(I)=255

C      INITIALIZE DISTRIBUTION CUT-OFF, PEROUT, AND
C      THE STANDARD DEVIATION MULTIPLE, LAM.
C
      LAM=2
      PEROUT=5
      DO 30 I=8,15
      30 COVHD1(I) = BLANK

C      DO 40 I=1,15
      40 COMENT(I) = BLANK

C      NOW SET UP REREAD BUFFER,
C
      CALL REREAD(30,80)
      50 COL=0
C      NOW READ A CARD INTO THE BUFFER
      READ(21,55)(ACARD(I),I=1,20)
      55 FORMAT(20A4)
      WRITE(30,55)(ACARD(I),I=1,20)
      REWIND 30
C      STATFILE CARD READ
      IF (NUSTAP .OR. NUSFIL) SCAFLG = 2
      READ (30,60)CODE,CARD2
      60 FORMAT(A4,6X,62A1)
      REWIND 30
      WRITE (6,70)CODE,CARD2
      70 FORMAT(T5,A4,6X,62A1)
      DO 80 I=1,CINMAX
      IF (CINDEX(I).EQ.CODE)GO TO(110,150,160,180,190,210,200,230,250,
      *270,360,280,290,500,380,430,480,490,600),I
      80 CONTINUE

C      90 WRITE (6,100)CODE,CARD2
C
      100 FORMAT(//// 5X,*** BAD CONTROL CARD - DATATR/SETUP8 *** // 5X,
      * A4, 6X, 62A1 /// )

C
      GO TO 50
C B-MATRIX CARD
      110 J=NXCHR(CARD2,COL)
      IF (J.EQ.BLANK) GO TO 540
      COL=COL-1
      M=FIND12(CARD2,COL,8TEST)
      IF (M.EQ.-1) GO TO 540
      HMTRIG=1
      IF (M.EQ.2) GO TO 120
C B-MATRIX DATA ON TAPE FILE
      KEY=2
C READ B-MATRIX ARRAY FROM TAPE FILE
      CALL BMFIL(BMAT,LCOMB,NOFEAT,FETVEC,KEY)
      GO TO 130
C B-MATRIX DATA HEAD FROM CARD FILE
      120 KEY=1
      CALL BMFIL(BMAT,LCOMB,NOFEAT,FETVEC,KEY)
      130 NOFET2=NOFEAT
      NOFET4=LCOMB
      DO 140 B=1,NOFEAT
      140 FETVC2(B)=FETVEC(B)
      GO TO 50
C FEATURE CARD
      150 CONTINUE
      GO TO 50
C FORMAT CARD
      160 CONTINUE
      170 M=FIND12(CARD2,COL,FRVEC1)
      IF (M.EQ.-1) GO TO 540
      KZ=FIND12(CARD2,COL,SINVEC)

```

```

SET00800
SET00810
SET00820
SET00830
SET00840
SET00850
SET00860
SET00870
SET00880
SET00890
SET00900
SET00910
SET00920
SET00930
SET00940
SET00950
SET00960
SET00970
SET00980
SET00990
SET01000
SET01010
SET01020
SET01030
SET01040
SET01050
SET01060
SET01070
SET01080
SET01090
SET01100
SET01110
SET01120
SET01130
SET01140
SET01150
SET01160
SET01170
SET01180
SET01190
SET01200
SET01210
SET01220
SET01230
SET01240
SET01250
SET01260
SET01270
SET01280
SET01290
SET01300
SET01310
SET01320
SET01330
SET01340
SET01350
SET01360
SET01370
SET01380
SET01390
SET01400
SET01410
SET01420
SET01430
SET01440
SET01450
SET01460
SET01470
SET01480
SET01490
SET01500
SET01510
SET01520
SET01530
SET01540
SET01550
SET01560
SET01570
SET01580

```

SET01590
SET01600
SET01610
SET01620
SET01630
SET01640
SET01650
SET01660
SET01670
SET01680
SET01690
SET01700
SET01710
SET01720
SET01730
SET01740
SET01750
SET01760
SET01770
SET01780
SET01790
SET01800
SET01810
SET01820
SET01830
SET01840
SET01850
SET01860
SET01870
SET01880
SET01890
SET01900
SET01910
SET01920
SET01930
SET01940
SET01950
SET01960
SET01970
SET01980
SET01990
SET02000
SET02010
SET02020
SET02030
SET02040
SET02050
SET02060
SET02070
SET02080
SET02090
SET02100
SET02110
SET02120
SET02130
SET02140
SET02150
SET02160
SET02170
SET02180
SET02190
SET02200
SET02210
SET02220
SET02230
SET02240
SET02250
SET02260
SET02270
SET02280
SET02290
SET02300
SET02310
SET02320
SET02330
SET02340
SET02350
SET02360
SET02370

FILE: SETUP8

ORIGINAL PAGE IS
OF POOR QUALITY

```
C      M = FIND12( CARD2, COL, SINVEC )
C      IF (M.EQ.2) GO TO 290
C      GO TO 50
C      IF M = 3, 'T' OR 'TRANSF'
C      310 TRANSF = 1
C      M = FIND12( CARD2, COL, SINVEC )
C      IF (M.EQ.2) GO TO 290
C      GO TO 50
C      IF M = 4, 'S' --- CHECK FOR 'SCAFAC'
C      320 J = NXTCHR ( CARD2, COL )
C      IF NEXT CHARACTER IS 'C' , ASSUME 'SCAFAC'
C      IF (J.NE.CBCD) GO TO 540
C      Z = FIND12( CARD2, COL, SINVEC )
C      IF (Z.EQ.3) GO TO 330
C      GO TO 540
C      SCALE FACTOR OPTION : READ SCALING PAIRS, CON AND MIN , INTO
C      CONMIN
C      330 SCAFLG = 3
C      340 Z = FIND12( CARD2, COL, UP )
C      IF (Z.NE.2) GO TO 50
C      NMN = FLTNUM ( CARD2, COL, CONMIN(NSF) , 2 )
C      IF (NMN.NE.2) GO TO 540
C      ADDNUM = NSF + 1
C      IF ((NSF+NMN).GT.31) GO TO 50
C      NSF = NSF + NMN
C      Z = FIND12( CARD2, COL, CP )
C      IF (Z.EQ.2) GO TO 340
C      GO TO 540
C      PUNCH OPTION
C      350 NPUN = 1
C      GO TO 290
C      MODULE STAT DECK
C      350 MK=NXTCHR(CARD2,COL)
C      IF (MK.NE.BTEST(3)) GO TO 370
C      SCAFLG = 2
C      GO TO 50
C      370 CALL CRDSTA(ARRAY, TOP)
C      SCAFLG = 2
```

SET02380
SET02390
SET02400
SET02410
SET02420
SET02430
SET02440
SET02450
SET02460
SET02470
SET02480
SET02490
SET02500
SET02510
SET02520
SET02530
SET02540
SET02550
SET02560
SET02570
SET02580
SET02590
SET02600
SET02610
SET02620
SET02630
SET02640
SET02650
SET02660
SET02670
SET02680
SET02690
SET02700
SET02710
SET02720
SET02730
SET02740
SET02750
SET02760
SET02770
SET02780
SET02790
SET02800
SET02810
SET02820
SET02830
SET02840
SET02850
SET02860
SET02870
SET02880
SET02890
SET02900
SET02910
SET02920
SET02930
SET02940
SET02950
SET02960
SET02970
SET02980
SET02990
SET03000
SET03010
SET03020
SET03030
SET03040
SET03050
SET03060
SET03070
SET03080
SET03090
SET03100
SET03110
SET03120
SET03130
SET03140
SET03150
SET03160

FILE: SETUP8

```

C          GO TO 50
C          DATAFILE POSITIONING CARD
C          380 IF (NUDTAP.AND.NUDFIL) GO TO 50
C              M = NXTCHR ( CARD2 , COL )
C              IF (M.EQ.BLANK) GO TO 50
C              IF (M .EQ. UHCD) GO TO 410
C              IF (M .EQ. FRCD) GO TO 420
C          390 WRITE (6,400)
C          400 FORMAT(///// 5X, ***** DATATR/SETUP8 ***** ERROR ON INPUT DATA
C              *FILE CARD --- CONTINUING TO PROCESS INPUT ***** ///// )
C              GO TO 50
C          410 J=FIN12(CARD2,COL,EQUVEC)
C              IF (J.EQ.-1) GO TO 390
C              M=NUMBER(CARD2,COL,DATAP,ZERO)
C              COL=COL-1
C              IF (M.NE.1) GO TO 390
C              NUDTAP = .TRUE.
C              GO TO 380
C          420 J=FIN12(CARD2,COL,EQUVEC)
C              IF (J.EQ.-1) GO TO 390
C              FILNO = NUMBER ( CARD2, COL, DATFIL, ZERO )
C              IF (FILNO.NE.1) GO TO 390
C              NUDFIL = .TRUE.
C              DATFIL=DATFIL-1
C              COL=COL-1
C              GO TO 380
C          STATFILE POSITIONING CARD
C          430 M=NXTCHR(CARD2,COL)
C              IF (M.EQ.BLANK) GO TO 50
C              IF (M .EQ. URCD) GO TO 460
C              IF (M .EQ. FRCD) GO TO 470
C          440 WRITE (6,450)
C          450 FORMAT( ///// 5X, ***** DATATR/SETUP8 ***** ERROR ON INPUT OR OUT
C              *PUT CARD --- CONTINUING TO PROCESS INPUT ***** /// )
C              NUSTAP = .FALSE.
C              NUSFIL = .FALSE.
C              GO TO 50
C          460 J=FIN12(CARD2,COL,EQUVEC)
C              IF (J.EQ.-1) GO TO 440
C              M=NUMBER(CARD2,COL,SAVTAP,ZERO)
C              COL=COL-1
C              IF (M.NE.1) GO TO 440
C          *****NUSTAP = .TRUE.
C              GO TO 430
C          470 J=FIN12(CARD2,COL,EQUVEC)
C              IF (J.EQ.-1) GO TO 440
C              FILST = NUMBER( CARD2, COL, STAFIL, ZERO )
C              IF (FILST.NE.1) GO TO 440
C          *****NUSFIL = .TRUE.
C              STAFIL=STAFIL-1
C              COL=COL-1
C              GO TO 430

```

SET03170
 SET03180
 SET03190
 SET03200
 SET03210
 SET03220
 SET03230
 SET03240
 SET03250
 SET03260
 SET03270
 SET03280
 SET03290
 SET03300
 SET03310
 SET03320
 SET03330
 SET03340
 SET03350
 SET03360
 SET03370
 SET03380
 SET03390
 SET03400
 SET03410
 SET03420
 SET03430
 SET03440
 SET03450
 SET03460
 SET03470
 SET03480
 SET03490
 SET03500
 SET03510
 SET03520
 SET03530
 SET03540
 SET03550
 SET03560
 SET03570
 SET03580
 SET03590
 SET03600
 SET03610
 SET03620
 SET03630
 SET03640
 SET03650
 SET03660
 SET03670
 SET03680
 SET03690
 SET03700
 SET03710
 SET03720
 SET03730
 SET03740
 SET03750
 SET03760
 SET03770
 SET03780
 SET03790
 SET03800
 SET03810
 SET03820
 SET03830
 SET03840
 SET03850
 SET03860
 SET03870
 SET03880
 SET03890
 SET03900
 SET03910
 SET03920
 SET03930
 SET03940
 SET03950

1
 2
 3
 4
 5
 6
 7
 8
 9
 10
 11
 12
 13
 14
 15
 16
 17
 18
 19
 20
 21
 22
 23
 24
 25
 26
 27
 28
 29
 30
 31
 32
 33
 34
 35
 36
 37
 38
 39
 40
 41
 42
 43
 44
 45
 46
 47
 48
 49
 50
 51
 52
 53
 54
 55
 56
 57
 58
 59
 60
 61
 62
 63
 64
 65
 66
 67
 68
 69
 70
 71
 72
 73
 74
 75
 76
 77
 78
 79
 80
 81
 82
 83
 84
 85
 86
 87
 88
 89
 90
 91
 92
 93
 94
 95
 96
 97
 98
 99
 100
 101
 102
 103
 104
 105
 106
 107
 108
 109
 110
 111
 112
 113
 114
 115
 116
 117
 118
 119
 120
 121
 122
 123
 124
 125
 126
 127
 128
 129
 130
 131
 132
 133
 134
 135
 136
 137
 138
 139
 140
 141
 142
 143
 144
 145
 146
 147
 148
 149
 150
 151
 152
 153
 154
 155
 156
 157
 158
 159
 160
 161
 162
 163
 164
 165
 166
 167
 168
 169
 170
 171
 172
 173
 174
 175
 176
 177
 178
 179
 180
 181
 182
 183
 184
 185
 186
 187
 188
 189
 190
 191
 192
 193
 194
 195
 196
 197
 198
 199
 200
 201
 202
 203
 204
 205
 206
 207
 208
 209
 210
 211
 212
 213
 214
 215
 216
 217
 218
 219
 220
 221
 222
 223
 224
 225
 226
 227
 228
 229
 230
 231
 232
 233
 234
 235
 236
 237
 238
 239
 240
 241
 242
 243
 244
 245
 246
 247
 248
 249
 250
 251
 252
 253
 254
 255
 256
 257
 258
 259
 260
 261
 262
 263
 264
 265
 266
 267
 268
 269
 270
 271
 272
 273
 274
 275
 276
 277
 278
 279
 280
 281
 282
 283
 284
 285
 286
 287
 288
 289
 290
 291
 292
 293
 294
 295
 296
 297
 298
 299
 300
 301
 302
 303
 304
 305
 306
 307
 308
 309
 310
 311
 312
 313
 314
 315
 316
 317
 318
 319
 320
 321
 322
 323
 324
 325
 326
 327
 328
 329
 330
 331
 332
 333
 334
 335
 336
 337
 338
 339
 340
 341
 342
 343
 344
 345
 346
 347
 348
 349
 350
 351
 352
 353
 354
 355
 356
 357
 358
 359
 360
 361
 362
 363
 364
 365
 366
 367
 368
 369
 370
 371
 372
 373
 374
 375
 376
 377
 378
 379
 380
 381
 382
 383
 384
 385
 386
 387
 388
 389
 390
 391
 392
 393
 394
 395
 396
 397
 398
 399
 400
 401
 402
 403
 404
 405
 406
 407
 408
 409
 410
 411
 412
 413
 414
 415
 416
 417
 418
 419
 420
 421
 422
 423
 424
 425
 426
 427
 428
 429
 430
 431
 432
 433
 434
 435
 436
 437
 438
 439
 440
 441
 442
 443
 444
 445
 446
 447
 448
 449
 450
 451
 452
 453
 454
 455
 456
 457
 458
 459
 460
 461
 462
 463
 464
 465
 466
 467
 468
 469
 470
 471
 472
 473
 474
 475
 476
 477
 478
 479
 480
 481
 482
 483
 484
 485
 486
 487
 488
 489
 490
 491
 492
 493
 494
 495
 496
 497
 498
 499
 500
 501
 502
 503
 504
 505
 506
 507
 508
 509
 510
 511
 512
 513
 514
 515
 516
 517
 518
 519
 520
 521
 522
 523
 524
 525

480 CONTINUE
PESCAL = 1
GO TO 50

```

600      M = NATCHR(CARD2,COL)
        IF (M.EQ.BLANK) GO TO 50
        IF (M.EQ.UHCD) GO TO 660
        IF (M.EQ.FBCD) GO TO 670
660      J = FIND12(CARD2,COL,EQUVEC)
        IF (J.EQ.-1) GO TO 440
        M = NUMBER(CARD2,COL,TRFORM,ZERO)
        COL = COL - 1
        IF (M.NE.1) GO TO 440
        GO TO 600
670      J = FIND12(CARD2,COL,EQUVEC)
        IF (J.EQ.-1) GO TO 440
        M = NUMBER(CARD2,COL,NF,ZERO)
        COL = COL - 1
        GO TO 50

```

```

490  CONTINUE
      NR = NXTCHR(CARD2, COL)
      IF (NR.EQ.BLANK) GO TO 540
      COL = COL - 1
      VECMAX = 16 - NBS
      NK = NBS + 1
      NBS = FLTNUM (CARD2, COL, BIAS(NK), VECMAX)
      GO TO 50

```

IF RESCALING BY THE STATISTICAL METHOD, READ STATISTICS FROM
FILE (SAVTAP), REDUCE THE STATISTICS TO THE SET OF CHANNELS
SPECIFIED IN FEIVC2 , AND STORE IN ARRAY .

```
IF (ORIG.EQ.0) GO TO 530
```

```
CALL PRTCOV (ARRAY (COVAR2), ARRAY (AVAR2), VARSZ2, NOFET2, ARRAY (SUBDS2)
```

PRINT OUT THE INPUT TRANSFORMATION MATRIX

RETURN

~~13-21~~
257

FILE: SETUP8

```
C 540 WRITE (6,550)CODE,CARD2
C 550 FORMAT(//// SX,***** INVALID CONTROL CARD REJECTED BY DATATH/SET
C      *UP8 ***** // SX, A*,6X, 62A1 //// )
C
      GO TO 50
      END
```

SET04750
SET04760
SET04770
SET04780
SET04790
SET04800
SET04810
SET04820
SET04830
SET04840

```

C C C SUBROUTINE TRANSF
      * (XT, RMAT, IDATA, TOP, IL, K, LCOMB, NSAMP, BIAS)
      IMPLICIT INTEGER (A-Z)
      REAL XT(16), RMAT(480), BIAS(16)

      INCLUDE COMAK9.LIST
C DATA TRANSFORMATION COMMON BLOCK
      COMMON/TRBLCK/OUTFMF,NOFEAT,FLOINF(6), FETVEC(30)
C$END

      DIMENSION IDATA(TOP)

      SUBROUTINE TRANSF DOES A DATA-TRANSFORMATION USING THE
FORMULA
      XT = IDATA * RMAT * BIAS
      XT = COMPONENT(IL), TRANSFORMED DATA VECTOR
      IDATA = INPUT DATA VECTOR ( NOFEAT X 1 )
      RMAT = TRANSFORMATION MATRIX ( LCOMB X NOFEAT )
      BIAS = ADDITIVE BIAS

      DO 10 IT=1,NOFEAT
          JSAMP = (IT - 1) * NSAMP + K
          ZCOMB = LCOMB * (IT - 1) + IL
          XT(IL) = XT(IL) +
              IDATA(JSAMP) * RMAT(ZCOMB)
C
10 CONTINUE
      XT(IL) = XT(IL) + BIAS(IL)
      RETURN
      END

```

~~13-23~~
259

FILE: TRHIST

```

SUBROUTINE TRHIST(IDATA,AMAX,AMIN,ACON,EMAT,LCOMB,
*PENOUT,FILE,HIS,TOP,LAR,FLDNAM,NC,VERTCS,MAX,MIN,CON,
BIAS)
C HISTOGRAM THE TRANSFORMED DATA AND CALCULATE THE MIN MAX AND RANGE
C FOR THIS DATA TO ALLOW RESCALING IN THE 0-255 RANGE
C
C IMPLICIT INTEGER(A-Z)
C
C REAL BIAS(16), XT(16), PERCEN(16), MIN(16), MAX(16), CON(16)
C
C REAL EMAT(400), AMIN(16), AMAX(16), ACON(16)
C
C REAL XMIN, XPER, SUMFIL, DUMMY
C
C INCLUDE COM-K1.LIST
C INCLUDE COM-K9.LIST
C INCLUDE COM-K4.LIST
C INCLUDE COM-K5.LIST
C
COMMON/INFORM/NOCLS2,NOSUM2,NOFET2,VARSZ2,TOTVT2,NOFLD2,
AVAR2,COVAR2,CLS102,SUBNO2,SUMDS2,FLDSV2,VERTX2,
FFTVC2(30),SUFVC2(75),SUBPTR(75),CLSV2(60),
KEPPTS(60),NOGRP,GRPNAM(60),GRPDEX(61),
GRPCHK(61),GRPHPS(124)
C
C DIMENSION HED1(15),HED2(15),DATE(3),COMENT(15)
C EQUIVALENCE (HED1(1),HED1(4)),(DATE(1),HED1(22)),
2 (HED2(1),HED2(3)),(COMENT(1),HED2(48))
C COMMON/GLOBAL/HEAD(63),HAPTAP,DATE,SAVTAP,HMFILE,HMKEY,
HISFIL,HISKEY,INFORM,EXPTM,EXPKY,MAPUNT,NOFILE,
DRUMAD,DRUMDS,PAGSIZ,DATFIL,STAFIL,ASAV,ASAVL,
NHSTUN,NHSTFI,SCOUN,MAPFIL,
DOTUNT,DOTFIL,NCHPAS,TRNSFL,HMTRFL,HISTFL,PCHUNT,
CRDUNT,DOTUNT,WANDIO
C DATA TRANSFORMATION COMMON BLOCK
COMMON/TRPLCK/OUTFMT,NOFEAT,FLDINF(6), FETVEC(30)
CSEND
C DIMENSION FILHIS(LCOMB,101), IDATA(TOP), TOTPTS(16),
* VERTCS(2,11),FL(8),FLDINP(6)
C
C READ THE COORDINATES ( VERTICES ) OF THE FIELD, FOR THE
C DATA TO BE TRANSFORMED
C
10 LAR=LAR*AD(FLDNAM,VERTCS,FLDINF,NC)
IF (LAR.EQ.0) GO TO 210
IF (LAR.LE.-1) GO TO 10
C
C POSITION THE INPUT DATA TAPE AND READ IN THE HEADER RECORD
C
CALL TAPHOR( DATE , DATFIL )
C
DO 20 I=1,LCOMB
MAX(I)=AMAX(I)
MIN(I)=AMIN(I)
20 CON(I)=ACON(I)
NSAMP=(FLDINF(5)-FLDINF(4))/FLDINF(6)+1
LINES=(FLDINF(2)-FLDINF(1))/FLDINF(3)+1
DUMMY = (LINES*NSAMP)/2000
ALP=SORT(DUMMY)
IF (ALP.LE.1) ALP=1
FLDINP(1)=FLDINF(1)
FLDINP(2)=FLDINF(2)
FLDINP(4)=FLDINF(4)
FLDINP(5)=FLDINF(5)
FLDINP(3)=ALP
FLDINP(6)=ALP
LINES=(FLDINP(2)-FLDINP(1))/FLDINP(3)+1
NSAMP=(FLDINP(5)-FLDINP(4))/FLDINP(6)+1
CALL FLDIRT(FLDINP,FETVEC,NOFEAT)
DO 30 I=1,LCOMB
TOTPTS(I) = 0
DO 30 J=1,101
30 FILHIS(I,J)=0
DO 130 I=1,LINES
CALL LINFIL(IDATA,ENDTAP)
IF (ENDTAP.NE.0) GO TO 140

```

TRH00010
 TRH00020
 TRH00030
 TRH00040
 TRH00050
 TRH00060
 TRH00070
 TRH00080
 TRH00090
 TRH00100
 TRH00110
 TRH00120
 TRH00130
 TRH00140
 TRH00150
 TRH00160
 TRH00170
 TRH00180
 TRH00190
 TRH00200
 TRH00210
 TRH00220
 TRH00230
 TRH00240
 TRH00250
 TRH00260
 TRH00270
 TRH00280
 TRH00290
 TRH00300
 TRH00310
 TRH00320
 TRH00330
 TRH00340
 TRH00350
 TRH00360
 TRH00370
 TRH00380
 TRH00390
 TRH00400
 TRH00410
 TRH00420
 TRH00430
 TRH00440
 TRH00450
 TRH00460
 TRH00470
 TRH00480
 TRH00490
 TRH00500
 TRH00510
 TRH00520
 TRH00530
 TRH00540
 TRH00550
 TRH00560
 TRH00570
 TRH00580
 TRH00590
 TRH00600
 TRH00610
 TRH00620
 TRH00630
 TRH00640
 TRH00650
 TRH00660
 TRH00670
 TRH00680
 TRH00690
 TRH00700
 TRH00710
 TRH00720
 TRH00730
 TRH00740
 TRH00750
 TRH00760
 TRH00770
 TRH00780
 TRH00790

FILE: TRHIST

```

IF (I.NE.1) GO TO 40
ILIN=FLDINP(1)
GO TO 50
40 ILIN=ILIN+FLDINP(3)
50 CONTINUE
CALL FDLINT(VERTCS,NC,FL,ILIN,NS,JJ)
DO 110 K=1,NSAMP
KP=(K-1)*FLDINP(6)+FLDINP(4)
DO 100 LK=1,JJ,2
LKPI = LK + 1
IF (KP.LT.FL(LK)) GO TO 110
IF (KP.GT.FL(LKPI)) GO TO 90
DO 80 J=1.LCOMB
XT(J)=0.

CALL TRANSF TO DO A DATA TRANSFORMATION

CALL TRANSF
C (XT, BMAT, IDATA, TOP, J, K, LCOMB, NSAMP, BIAS)

HISTOGRAM THE TRANSFORMED DATA ( USING TRANSFORMED DATA MAX
AND MIN AND SCALE FACTOR, CON, COMPUTED IN SURR. MAXMAT
TO OBTAIN THE HISTOGRAM "BIN LEVEL" FOR EACH TRANSFORMED
DATA POINT )

IF (XT(J).LE.MIN(J)) GO TO 60
IF (XT(J).GE.MAX(J)) GO TO 70
DPT=(XT(J)-MIN(J))/CON(J)+1

IF ( DPT .LE. 0 ) DPT = 1
IF ( DPT .GT. 101 ) DPT = 101

FILHIS(J,DPT)=FILHIS(J,DPT)+1
GO TO 80
60 FILHIS(J,1)=FILHIS(J,1)+1
GO TO 80
70 FILHIS(J,101) = FILHIS(J,101) + 1
80 TOTPTS(J)=TOTPTS(J)+1
GO TO 110
90 IF (LKPI.GE.JJ) GO TO 120
100 CONTINUE
110 CONTINUE
120 CONTINUE
130 CONTINUE
140 CONTINUE

ELIMINATE PEROUT/2 OF POINTS FROM UPPER AND LOWER TAILS OF
THE TRANSFORMED DATA-DISTRIBUTION --- OBTAIN THE REVISED MAX
AND SCALING PARAMETERS CON AND MIN AFTER APPLICATION OF PEROUT

XPER=PEROUT/200.
C CALCULATE MIN,MAX,AND CON ARRAYS
DO 150 I=1.LCOMB
150 PERCEN(I)=XPER*TOTPTS(I)
DO 200 J=1.LCOMB
XMIN=MIN(I)
SUMFIL=0.
DO 160 J=1,101
SUMFIL=SUMFIL+FILHIS(I,J)
IF (SUMFIL.LT.PERCEN(I)) GO TO 160
MIN(I)=(J-1)*CON(I)+XMIN
GO TO 170
160 CONTINUE
170 SUMFIL=0.
J=102
175 J=J-1
SUMFIL=SUMFIL+FILHIS(I,J)
IF (SUMFIL.LT.PERCEN(I)) GO TO 180
JMI = J - 1
MAX(I) = JMI * CON(I) + XMIN
GO TO 190
180 IF (J.GT.1) GO TO 175
190 CONTINUE
CON(I) = 255. / (MAX(I) - MIN(I))
200 CONTINUE

```

ORIGINAL PAGE IS
OF POOR QUALITY

TRH00800
 TRH00810
 TRH00820
 TRH00830
 TRH00840
 TRH00850
 TRH00860
 TRH00870
 TRH00880
 TRH00890
 TRH00900
 TRH00910
 TRH00920
 TRH00930
 TRH00940
 TRH00950
 TRH00960
 TRH00970
 TRH00980
 TRH00990
 TRH01000
 TRH01010
 TRH01020
 TRH01030
 TRH01040
 TRH01050
 TRH01060
 TRH01070
 TRH01080
 TRH01090
 TRH01100
 TRH01110
 TRH01120
 TRH01130
 TRH01140
 TRH01150
 TRH01160
 TRH01170
 TRH01180
 TRH01190
 TRH01200
 TRH01210
 TRH01220
 TRH01230
 TRH01240
 TRH01250
 TRH01260
 TRH01270
 TRH01280
 TRH01290
 TRH01300
 TRH01310
 TRH01320
 TRH01330
 TRH01340
 TRH01350
 TRH01360
 TRH01370
 TRH01380
 TRH01390
 TRH01400
 TRH01410
 TRH01420
 TRH01430
 TRH01440
 TRH01450
 TRH01460
 TRH01470
 TRH01480
 TRH01490
 TRH01500
 TRH01510
 TRH01520
 TRH01530
 TRH01540
 TRH01550
 TRH01560
 TRH01570
 TRH01580

FILE: TRMIST

C
C
C
C
C
C

RETURN THE SCALING PARAMETERS. CON AND MIN , REQUIRED TO
RESCALE THE TRANSFORMED DATA TO THE RANGE , 0 - 255 .

210 RETURN
END

TRH01590
TRH01600
TRH01610
TRH01620
TRH01630
TRH01640
TRH01650
TRH01660

14. TRSTAT PROCESSOR

FILE: TRSTAT

| | | |
|-------|--|----------|
| | SUBROUTINE TRSTAT(ARRAY, TOP) | TRS00010 |
| | IMPLICIT INTEGER(A-Z) | TRS00020 |
| | DIMENSION ARRAY(TOP) | TPS00030 |
| C | INCLUDE COM4K1.LIST | TPS00040 |
| C | INCLUDE COM4K6.LIST | TPS00050 |
| | COMMON/INFORM/NOCLS2, NOSUR2, NOFET2, VARSZ2, TOTVT2, NOFLD2, | COM00010 |
| | AVAR2, COVAR2, CLS102, SUBNO2, SURDS2, FLOSV2, VERTX2, | COM00020 |
| | FETVC2(30), SUBVC2(75), SURPTH(75), CLSVC2(60), | COM00030 |
| | PERPTS(60), NOGRP, GRPNAM(60), GRPDEX(61), | COM00040 |
| | GRPCHK(61), GROUPS(124) | COM00050 |
| | COMMON/GLORAL/HEAD(63), MAPTAP, DATAPE, SAVTAP, BMFILE, BMKEY, | COM20010 |
| | HISFIL, HISKEY, TRFORM, EPIPTP, ERPKEY, MAPUNT, NOFILF, | COM40020 |
| | DRUMAD, DRUMDS, PAGESIZ, DATFIL, STAFIL, ASAV, ASAVFL | COM00030 |
| | , NHSTUN, NHSTFI, SCTRUN, MAPFIL | COM00040 |
| | , DOTUNT, DOTFIL, NCHPAS, TRNSFL, RMTFL, HISTFL, PCHUNT, | COM00050 |
| | CRDUNT, PRUNT, RANDIO | COM00060 |
| CSEND | REAL AMAT(1800), R(30) | TRS00070 |
| | CALL SETUP4(ARRAY, TOP, AMAT, ROW, IP, TRAN, B) | TPS00080 |
| | CALL TRMTX(ARRAY, TOP, AMAT, ROW, IP, TRAN, B) | TPS00090 |
| | RETURN | TPS00100 |
| | END | TPS00110 |
| | | TPS00120 |

ORIGINAL PAGE 1-
OF POOR QUALITY

FILE: AMFIL

```
SUBROUTINE AMFIL(ROW,COLUMN,AMAT,VEC,B)
IMPLICIT INTEGER(A-Z)
C INCLUDE COMMON
COMMON/GLOBAL/HEAD(63),MAYTAP,DATAPE,SAVTAP,BMFILE,BMKEY,
* HISFIL,HISKEY,TRFORM,ERIPTP,ERPKEY,MAPUNT,NOFILF,
* DRUMAD,DRUMWDS,PAGSIZ,DATFIL,STAFIL,ASAV,ASAVFL
* .NHSTUN,NHSTFI,SCTRUN,MAPFIL
* .DOTUNT,DOTFIL,NCHPAS,TRANSFL,BMTRFL,HISTFL,PCHUNT,
* CROUT,PRUNT,RANDIO
C$END
REAL AMAT(1),B(30)
DIMENSION VEC(1)
1 READ(2,2) ROW,COLUMN,(VEC(I),I=1,COLUMN)
IK=ROW*COLUMN
READ(2,3) (AMAT(I),I=1,IK)
READ(2,3) (B(I),I=1,ROW)
RETURN
2 FORMAT(5X,I2,5X,I2,2X,30I2)
3 FORMAT(5X,5E15.8)
END
```

AMF00010
AMF00020
AMF00030
AMF00040
AMF00050
AMF00060
AMF00070
AMF00080
AMF00090
AMF00100
AMF00110
AMF00120
AMF00130
AMF00140
AMF00150
AMF00160
AMF00170
AMF00180
AMF00190
AMF00200

FILE: AMFILE

| | | |
|-------|--|----------|
| C* | SURROUTINE AMFILE(ROW,NOCHAN,CHNVEC,AMAT,BVEC) | AMF00010 |
| C* | AMFILE WILL READ INTO CORE THE A-MATRIX AND B VECTOR FROM UNIT | AMF00020 |
| C* | 7 | AMF00030 |
| C* | IMPLICIT INTEGER (A-Z) | AMF00040 |
| | REAL AMAT(1),BVEC(1) | AMF00050 |
| C | INCLUDE COMRK6 | AMF00060 |
| | COMMON/GLOBAL/HEAD(63),MPTAP,DATAP,SAVTAP,RMFILE,RMKEY, | AMF00070 |
| | HISFIL,HISKEY,TRFORM,ERIPTP,ERPKEY,MAPUNT,NOFILE, | AMF00080 |
| * | DRUMAD,DRM=DS,PAGSIZ,DATFIL,STAFIL,ASAV,ASAVFL | AMF00090 |
| * | .NHSTUN,NHSTFI,SCTHUN,MAPFIL | AMF00100 |
| * | .DOTUNT,DOTFIL,NCHPAS,TRNSFL,BMTRFL,HISTFL,PCHUNT, | AMF00110 |
| * | CRDUNT,PRUNT,HANDIO | AMF00120 |
| CSEND | DIMENSION CHNVEC(1) | AMF00130 |
| | READ(7) ROW,NOCHAN,(CHNVEC(I),I=1,NOCHAN) | AMF00140 |
| | TOTAL = NOCHAN**2 | AMF00150 |
| C | READ(7) (AMAT(I),I=1,TOTAL),(BVEC(I),I=1,NOCHAN) | AMF00160 |
| | RETURN | AMF00170 |
| | END | AMF00180 |
| | | AMF00190 |
| | | AMF00200 |
| | | AMF00210 |
| | | AMF00220 |

SET00010
SET00020
SET00030
SET00040
SET00050
SET00060
SET00070
SET00080
SET00090
SET00100
SET00110
SET00120
SET00130
SET00140
SET00150
SET00160
SET00170
SET00180
SET00190
SET00200
SET00210
SET00220
SET00230
SET00240
SET00250
SET00260
SET00270
SET00280
SET00290
SET00300
SET00310
SET00320
SET00330
SET00340
SET00350
SET00360
SET00370
SET00380
SET00390
SET00400
SET00410
SET00420
SET00430
SET00440
SET00450
SET00460
SET00470
SET00480
SET00490
SET00500
SET00510
SET00520
SET00530
SET00540
SET00550
SET00560
SET00570
SET00580
SET00590
SET00600
SET00610
SET00620
SET00630
SET00640
SET00650
SET00660
SET00670
SET00680
SET00690
SET00700
SET00710
SET00720
SET00730
SET00740
SET00750
SET00760
SET00770
SET00780
SET00790

FILE: SETUP9

```

      6 NOSUB2=NUMBER(CARD,COL,SURVC2,NOSUB2)
      CALL ORDER(SURVC2,NOSUB2)
      GO TO 13
C READ IN A-MATRIX
      8 CONTINUE
      AMTXSW = -1
      CALL AMFIL(POW,NOCHAN,AMAT,CHNVEC,8)
      GO TO 13
C OPTIONS
      9 J=NXTCHR(CARD,COL)
      IF(J.EQ.BLANK) GO TO 16
      IF(J.EQ.P) IP=1
      IF(J.EQ.O) ORIG=1
      IF(J.EQ.T) TRAN=1
      M=FIN12(CARD,COL,SINVEC)
      IF(M.NE.2) GO TO 13
      GO TO 9
C MODULE CARD
      11 J=NXTCHR(CARD,COL)
      IF(J.EQ.F) GO TO 13
      CALL CRDSTA(ARRAY,TOP)
      GO TO 13
C*
C* STAT FILE CARD
C*
      20 M = NXTCHR(CARD,COL)
      21 IF (M.EQ.BLANK) GO TO 13
      IF (M.EQ.URCD) GO TO 25
      IF (M.EQ.FRCD) GO TO 30
      22 WRITE(4,222)
      222 FORMAT(' ERROR ON STAT FILE CARD')
      GO TO 13
      25 J = FIND12(CARD,COL,SLASH)
      IF (J.EQ.-1) GO TO 22
      23 M = NXTCHR(CARD,COL)
      IF (M.EQ.URCD) GO TO 26
      IF (M.EQ.FRCD) GO TO 28
      GO TO 21
      26 J = FIND12(CARD,COL,SINVEC)
      IF (J.NE.3) GO TO 22
      M = NUMBER(CARD,COL,SAVTAP,ZERO)
      COL = COL - 1
      GO TO 23
      28 J = FIND12(CARD,COL,SINVEC)
      IF (J.NE.3) GO TO 22
      M = NUMBER(CARD,COL,STAFIL,ZERO)
      STAFIL = STAFIL - 1
      COL = COL - 1
      GO TO 23
      30 J = FIND12(CARD,COL,SLASH)
      IF (J.EQ.-1) GO TO 22
      32 M = NXTCHR(CARD,COL)
      IF (M.EQ.URCD) GO TO 34
      IF (M.EQ.FRCD) GO TO 36
      GO TO 21
      34 J = FIND12(CARD,COL,SINVEC)
      IF (J.NE.3) GO TO 22
      M = NUMBER(CARD,COL,ASAV,ZERO)
      COL = COL - 1
      GO TO 32
      36 J = FIND12(CARD,COL,SINVEC)
      IF (J.NE.3) GO TO 22
      M = NUMBER(CARD,COL,ASAVFL,ZERO)
      ASAVFL = ASAVFL - 1
      COL = COL - 1
      GO TO 32
C*
C* CHANNEL CARD
C*
      40 M = NXTCHR(CARD,COL)
      IF (M.EQ.BLANK) GO TO 16
      COL = COL - 1
      NOFET2 = NUMBER(CARD,COL,FETVC2,NOFET2)
      CALL ORDER(FETVC2,NOFET2)
      GO TO 13
C *END* CARD
      60 CALL RFDOSAV(ARRAY,TOP,RMSWT)
      VARS72=NOFET2*(NOFET2+1)/2
      IF (AMTXSW.EQ.-1) GO TO 65

```

ORIGINAL PAGE IS
OF POOR QUALITY

ST00800
 ST00810
 ST00820
 ST00830
 ST00840
 ST00850
 ST00860
 ST00870
 ST00880
 ST00890
 ST00900
 ST00910
 ST00920
 ST00930
 ST00940
 ST00950
 ST00960
 ST00970
 ST00980
 ST00990
 ST01000
 ST01010
 ST01020
 ST01030
 ST01040
 ST01050
 ST01060
 ST01070
 ST01080
 ST01090
 ST01100
 ST01110
 ST01120
 ST01130
 ST01140
 ST01150
 ST01160
 ST01170
 ST01180
 ST01190
 ST01200
 ST01210
 ST01220
 ST01230
 ST01240
 ST01250
 ST01260
 ST01270
 ST01280
 ST01290
 ST01300
 ST01310
 ST01320
 ST01330
 ST01340
 ST01350
 ST01360
 ST01370
 ST01380
 ST01390
 ST01400
 ST01410
 ST01420
 ST01430
 ST01440
 ST01450
 ST01460
 ST01470
 ST01480
 ST01490
 ST01500
 ST01510
 ST01520
 ST01530
 ST01540
 ST01550
 ST01560
 ST01570
 ST01580

FILE: SETUP9

| | |
|--|----------|
| CALL AMFILE (ROW,NOCHAN,CHNVEC,AMAT,B) | SET01590 |
| 65 CONTINUE | SET01600 |
| C WRITE A-MATRIX | SET01610 |
| CALL WRTAMT (AMAT,ROW,NOFET2,FETVC2,B) | SET01620 |
| IF (ORIG.EQ.1) CALL PRTCOV (ARRAY (COVAR2),ARRAY (AVAR2),VARSZ2, | SET01630 |
| NOFET2,ARRAY (SUBDS2)) | SET01640 |
| * IF (NOFET2 .EQ. NOCHAN) GO TO 13 | SET01650 |
| WRITE (6,100)NOFET2,NOCHAN | SET01660 |
| 100 FORMAT(' NO. OF CHANNELS FROM STAT FILE DOES NOT EQUAL THE', | SET01670 |
| ' NO. OF CHANNELS ON A-MATRIX FILE. '// CHANNELS ON STAT FILE = ', | SET01680 |
| ' 12// CHANNELS ON A-MATRIX = ', 12) | SET01690 |
| CALL CMERR | SET01700 |
| C SEND* CARD | SET01710 |
| GO RETURN | SET01720 |
| 16 WRITE (6,19) CARD | SET01730 |
| 19 FORMAT(' INVALID CONTROL CARD REJECTED ***SETUP9***') | SET01740 |
| GO TO 13 | SET01750 |
| END | SET01760 |

FILE: TRAMTX

```

SUBROUTINE TRAMTX (ARRAY, TOP, AMAT, ROW, IP, TRAN, B)
IMPLICIT INTEGER (A-Z)
REAL C (900), B (30), AMAT (ROW, *NOFET2), AMEAN (1800),
*NO (15), CC (900)
C
C
C
INCLUDE COMRK1.LIST
INCLUDE COMRK4.LIST
INCLUDE COMRK6.LIST
COMMON / INFORM / NOCLS2, NOSUR2, NOFET2, VARSZ2, TOTVT2, NOFLD2,
* AVAR2, COVAR2, CLSID2, SUBNO2, SUBDS2, FLDSV2, VERTX2,
* FETVC2 (30), SUBVC2 (75), SUBPTR (75), CLSVC2 (60),
* KEPPTS (60), NOGRP, GRPNAM (60), GRPDEX (61),
* GRPCHK (61), GROUPS (124)
DIMENSION HED1 (15), HED2 (15), DATE (3), COMENT (15)
EQUIVALENCE (HED1 (1), HEAD (4)), (DATE (1), HEAD (22)),
2 (HED2 (1), HEAD (30)), (COMENT (1), HEAD (48))
COMMON / GLOBAL / HEAD (63), MAPTAP, DATAPE, SAVTAP, RMFILE, BMKEY,
* HISFIL, HISKEY, TRFORM, ERIPTR, ERPKEY, MAPUNT, NOFILE,
* DRUMAD, DRUMDS, PAGESIZ, DATFIL, STAFIL, ASAV, ASAVFL
* NMSTUN, NMSTFI, SCTRUN, MAPFIL
* DOTUNT, DOTFIL, NCHPAS, TRANSFL, BMTRFL, HISTFL, PCHUNT,
* CRDUNT, PRUNT, RANDIO
CSENO
DIMENSION ARRAY (TOP), NSUB (60)
PUNCH = PCHUNT
NO 41 I = 1, NOSUR2
TPP = SUBDS2 + 1 - 1
NSUB (I) = ARRAY (IPP)
NOFLD = NOFLD2
REWIND ASAV
C
C
C
IF (ASAVFL .EQ. 0) GO TO 100
POSITION STAT FILE
CALL FSRFL (ASAV, ASAVFL, ISTAT)
IF (ISTAT .EQ. 0) GO TO 100
FILNO = ASAVFL + 1
WRITE (6, 110) FILNO
110 FORMAT (' ERROR IN TRYING TO POSITION TRANSFORMED STAT FILE TO BEGI
*NING OF FILE ', I3)
CALL CMERR
100 CONTINUE
C
WRITE (ASAV) NOCLS2, NOSUR2, POW, NOFLD, TOTVT2,
* (FETVC2 (I), I = 1, NOFET2)
IF (IP .NE. 1) GO TO 38
WRITE (PUNCH, 33)
33 FORMAT (' MODULE TRAINING FIELD DECK ')
WRITE (PUNCH, 34) NOCLS2, NOSUR2, ROW, NOFLD, TOTVT2
34 FORMAT (' NOCLS ', I4, ' NOSUR ', I2, ' NOFEAT ', I2, ' NOFLD ', I3,
* ' TOTVT ', I4)
WRITE (PUNCH, 35) (FETVC2 (I), I = 1, NOFET2)
35 FORMAT (' CHNVEC ', 4X, 30I2)
38 CONTINUE
J = FLDSV2
K = VERTX2
1 NO 2 I = 1, NOFLD
JJ = J + 3
KK = K + 2 * ARRAY (JJ) - 1
WRITE (ASAV) (ARRAY (N), N = J, JJ)
WRITE (ASAV) (ARRAY (N), N = K, KK)
IF (IP .NE. 1) GO TO 39
WRITE (PUNCH, 36) (ARRAY (N), N = J, JJ)
36 FORMAT (' A4, 4X, I2, RX, I2, RX, I2)
WRITE (PUNCH, 37) (ARRAY (N), N = K, KK)
37 FORMAT (' VERTICES ', I4I5)
39 CONTINUE
J = JJ + 1
K = KK + 1
2 CONTINUE
KK = SUBNO2 + NOCLS2 - 1
LL = SUBDS2 + NOSUR2 - 1
WRITE (ASAV) (ARRAY (J), J = 1, NOCLS2), (ARRAY (K), K = SUBNO2, KK),
* (ARRAY (L), L = SUBDS2, LL)
IF (IP .EQ. 0) GO TO 23
WRITE (PUNCH, 31) (ARRAY (J), J = 1, NOCLS2)
31 FORMAT (' CLSIDF ', 2X, A4, 4X, I4I4)
WRITE (PUNCH, 32) (ARRAY (K), K = SUBNO2, KK)
32 FORMAT (' SUBNO ', 24 (1X, I2))

```

ORIGINAL PAGE IS
OF POOR QUALITY

TRA00010
 TRA00020
 TRA00030
 TRA00040
 TRA00050
 TRA00060
 TRA00070
 TRA00080
 TRA00090
 TRA00100
 TRA00110
 TRA00120
 TRA00130
 TRA00140
 TRA00150
 TRA00160
 TRA00170
 TRA00180
 TRA00190
 TRA00200
 TRA00210
 TRA00220
 TRA00230
 TRA00240
 TRA00250
 TRA00260
 TRA00270
 TRA00280
 TRA00290
 TRA00300
 TRA00310
 TRA00320
 TRA00330
 TRA00340
 TRA00350
 TRA00360
 TRA00370
 TRA00380
 TRA00390
 TRA00400
 TRA00410
 TRA00420
 TRA00430
 TRA00440
 TRA00450
 TRA00460
 TRA00470
 TRA00480
 TRA00490
 TRA00500
 TRA00510
 TRA00520
 TRA00530
 TRA00540
 TRA00550
 TRA00560
 TRA00570
 TRA00580
 TRA00590
 TRA00600
 TRA00610
 TRA00620
 TRA00630
 TRA00640
 TRA00650
 TRA00660
 TRA00670
 TRA00680
 TRA00690
 TRA00700
 TRA00710
 TRA00720
 TRA00730
 TRA00740
 TRA00750
 TRA00760
 TRA00770
 TRA00780
 TRA00790

FILE: TRAMTX

```

WRITE(PUNCH,40) (ARRAY(L),L=SUBDS2,LL)
40 FORMAT((SUBDES ',10(A4,3X)))
23 CONTINUE
C FOR EACH SUBCLASS COMPUTE TRANS STATS AND OUTPUT THEM
J=JVAR2
K=COVAR2
L=1
KKK=1
DO 20 I=1,NOSUB2
  RR=ROW*(ROW+1)/2
  P=SUBVC2(I)
  KCNT=KKK+RR-1
C MULTIPLY A-MATRIX BY MEAN VECTOR
  CALL MATVEC(AMAT,ARRAY(J),AMEAN(L),ROW,NOFET2)
C ADD P VECTOR TO GET TRANSFORMED MEANS
  KM=0
  LL=L+ROW-1
  DO 21 I21=L,LL
    KM=KM+1
    21 AMFAN(I21)=AMEAN(I21)+8(KM)
C COMPUTE TRANSFORMED COVAR MATRIX
C A-MATR * COVAR * A-MATH TRANSPOSE
C MULTIPLY A-MATR * COVAR AND STORE IN C
  CALL MTMLS6(AMAT,ARRAY(K),C,ROW,NOFET2)
C MULTIPLY C BY A TRANSPOSE AND STORE IN ARRAY
  CALL MTMDAT(C,AMAT,CC,ROW,NOFET2,ROW,OD,ARRAY(KKK))
  WRITE(ASAV) KEPPTS(P),(ARRAY(I21),I21=KKK,KCNT),
    (AMEAN(I21),I21=L,LL)
  IF(IP.FO.0) GO TO 22
  WRITE(PUNCH,95) KEPPTS(P)
95 FORMAT('NOPTS ',6X,I4)
  WRITE(PUNCH,96) (AMEAN(I21),I21=L,LL)
96 FORMAT('MEANS ',5E15.8)
  WRITE(PUNCH,97) (ARRAY(I21),I21=KKK,KCNT)
97 FORMAT('COVAR ',5E15.8)
22 CONTINUE
  J=J+ROW
  L=LL+1
  CCC=CCC+RR
  K=K+VARS22
20 CONTINUE
IF(TPAN.NE.1) RETURN
CV1=ROW*(ROW+1)/2
CALL PRTCOV(ARRAY(1),AMEAN(1),CV1,ROW,NSUB(1))
ENDFILE ASAV
PEWIND ASAV
END

```

TRA00800
 TRA00810
 TRA00820
 TRA00830
 TRA00840
 TRA00850
 TRA00860
 TRA00870
 TRA00880
 TRA00890
 TRA00900
 TRA00910
 TRA00920
 TRA00930
 TRA00940
 TRA00950
 TRA00960
 TRA00970
 TRA00980
 TRA00990
 TRA01000
 TRA01010
 TRA01020
 TRA01030
 TRA01040
 TRA01050
 TRA01060
 TRA01070
 TRA01080
 TRA01090
 TRA01100
 TRA01110
 TRA01120
 TRA01130
 TRA01140
 TRA01150
 TRA01160
 TRA01170
 TRA01180
 TRA01190
 TRA01200
 TRA01210
 TRA01220
 TRA01230
 TRA01240
 TRA01250
 TRA01260

FILE: WRTAMT

```

SUBROUTINE WRTAMT (AMAT, ROW, COLUMN, FETVC2, H)
IMPLICIT INTEGER (A-Z)
C INCLUDE COM-K4.LIST
C INCLUDE COM-K4.LIST
DIMENSION HEAD(15), HEAD2(15), DATE(3), COMMENT(15)
EQUIVALENCE (HEAD(1), HEAD(4)), (DATE(1), HEAD(22)),
2 (HEAD(1), HEAD(30)), (COMMENT(1), HEAD(48))
COMMON/GLOBAL/HEAD(63), WRTAMT, DATE, SAVTAM, HMFIL, HMKY,
• HISEIL, HISKFY, TRFORM, EN1TP, FADKEY, MAPUNT, NOFILE,
• DRIMAN, DRMWS, BAGS17, DATEIL, STAFIL, ASAV, ASAVFL
• NHSTUN, INSTFI, SCTRUN, MAPFIL
• DOTUNT, DOTFIL, JCHPAS, TRNSFL, HMTXFL, HISTFL, PCHUNT,
• CPDUNT, PCHUNT, RANDIO
CSEND
REAL AMAT(ROW, COLUMN), R(1)
DIMENSION FETVC2(1), CH(2)
DATA CH/CH(1,1), CH(1,2)
WRITE (6, HEAD)
WRITE (6, HEAD2)
11 FORMAT(1, 1, VECTOR = 1, //10X, 12F10.4/10X, 12E10.4/10X, 6E10.4)
WRITE (6, 1) FOR COLUMN
IR=1
IK=12
5 IF (IK.GT. COLUMN) IK= COLUMN
WRITE (6, 2) (CH, I=1, IK)
WRITE (6, 3) (FETVC2(I), I=IR, IK)
WRITE (6, 4)
DO 10 J=1, ROW
10 WRITE (6, 4) J, (AMAT(I, I), I=1, IK)
IF (IK.EQ. COLUMN) RETURN
IR=IR+1
IK=IK+12
GO TO 5
1 FORMAT(///57X, 'A-MATRIX'//
• 50X, 'NO. OF COMBINATIONS -', I3/
• 50X, 'NO. OF CHANNELS -', I3)
2 FORMAT(//12X, 12(2A4, 2X))
3 FORMAT(14X, 11(12, 2X), 12)
6 FORMAT(' COMBINATIONS')
4 FORMAT(1X, 15, 4X, 12(1X, E4, 3))
END

```

WRT00010
WRT00020
WRT00030
WRT00040
WRT00050
WRT00060
WRT00070
WRT00080
WRT00090
WRT00100
WRT00110
WRT00120
WRT00130
WRT00140
WRT00150
WRT00160
WRT00170
WRT00180
WRT00190
WRT00200
WRT00210
WRT00220
WRT00230
WRT00240
WRT00250
WRT00260
WRT00270
WRT00280
WRT00290
WRT00300
WRT00310
WRT00320
WRT00330
WRT00340
WRT00350
WRT00360
WRT00370
WRT00380
WRT00390
WRT00400
WRT00410

ORIGINAL PAGE IS
OF POOR QUALITY

15. NDHIST PROCESSOR

FILE: NDHIST

```
C SUBROUTINE NDHIST(ARRAY, TOP)
C NDHIST IS THE DRIVER ROUTINE FOR THE N-DIMENSIONAL PROCESSOR
C TWO LARGE ARRAYS ARE USED - PLANK COMMON ARRAY CALLED ARRAY(TOP)
C AND AN ARRAY CALLED HIST(LIMIT)
C DATA LIMIT/12000/
C DIMENSION HIST(12000)
C DIMENSION ARRAY(1).
C READ IN CONTROL CARDS
C CALL SETIO(LIMIT)
C NDHIST IS THE ORGANIZER
C CALL NDHST1(HIST, ARRAY(1), ARRAY(2801), ARRAY(801),
C   LIMIT, ARRAY(1), TOP)
C RETURN
C END
```

CH

ORIGINAL PAGE IS
OF POOR QUALITY

FILE: ADDRES

```

SUBROUTINE ADDRES(TOP,NSAMP,NOFEAT,REGIN,REGIN1)
  IMPLICIT INTEGER (A-Z)
  INCLUDE CMH11.LIST
  INCLUDE CMH11.LIST
  COMMON /NDIM/NCLCH,CLRVEC(30),MAXVEC,MAPKEY,
    CLASS,SURCLS,FIELD,MEANSW,NOVEC,FLDINF(6),SIZE,TOTMNS,
    CNTN1,CNTN2,ID1,ID2,COLOR1,COLOR2,HUFLEN,ID3,COLOR3,NODUMP,
    IDATA,TOTVEC
  COMMON BLOCK NDIM,IS USED ONLY BY THE N-DIMENSIONAL HISTOGRAM
    PROCESSOR
  DEFINITIONS
  NCLCH - NO. OF COLOR CHANNELS
  CLRVEC - ARRAY CONTAINING COLOR CHANNELS
  MAXVEC - MAXIMUM NO. OF VECTORS ARRAY HIST CAN STORE
  CLASS - KEY INDICATING FIELDS WILL BE GROUPED ON CLASS BASES
  SURCLS - KEY INDICATING FIELDS WILL BE GROUPED ON SURCLASS BASES
  FIELD - KEY INDICATING FIELDS WILL BE GROUPED ON FIELD BASES
  MEANSW - KEY INDICATING MEANS FOR INPUT FIELDS WILL BE COMPUTED
  NOVEC - NO. OF UNIQUE VECTOR HISTOGRAMMED
  FLDINF - ARRAY CONTAINING RECTANGULAR FIELD COORDINATES AROUND THE
    INPUT FIELDS
  SIZE - NOFEAT/4 - NO. OF COMPUTER WORDS TO STORE A PACKED
    HISTOGRAMMED VECTOR
  TOTMNS - TOTAL NO. OF ELEMENTS IN ARRAY CONTAINING MEANS
  CNTN2 - BEGINNING DRUM ADDRESS FOR STORING FREQUENCY
  ID1 - ADDRESS FOR STORING ID CODES IN ARRAY
  ID2 - BEGINNING DRUM ADDRESS FOR STORING ID CODES
  COLOR1 - ADDRESS FOR STORING COLOR CODES IN ARRAY
  COLOR2 - BEGINNING DRUM ADDRESS FOR STORING COLOR CODES
  HUFLEN - AMOUNT OF STORAGE AVAILABLE FOR STORING ID/COLOR CODES
  ID3 - ACCUMULATIVE ID CODE DRUM ADDRESS
  COLOR3 - ACCUMULATIVE COLOR CODE DRUM ADDRESS
  NODUMP - NO OF TIMES ID/COLOR CODES WERE DUMPED ON DRUM
  IDATA - ADDRESS FOR STORING IMAGERY DATA IN ARRAY
  TOTVEC - TOTAL NO. OF VECTORS IN THE AREA HISTOGRAMMED
  CONTINUE
  C GRAYMAP AND HIST COMMON BLOCK
  CSEND
  BASE ADDRESSES FOR ARRAY
  ID1 = 350
  IDATA = TOP - (NSAMP*NOFEAT + 1)
  IF (IDATA.GT. 6400) GO TO 110
  WRITE(6,100)
  100 FORMAT(' TOO MUCH DATA REQUESTED. REDUCE NO. OF SAMPLES PER SCAN
    * LINE// AND/OR NO. OF CHANNELS')
  CALL CMH11
  110 CONTINUE
  REMD = 3000
  IF (NCLCH.NE. 0) HUFLEN = REMD / 2
  IF (NCLCH.EQ. 0) HUFLEN = REMD
  IF (MEANSW.EQ. 1) HUFLEN = REMD
  COLOR1 = ID1 + HUFLEN
  DRUM ADDRESSES
  CNTN2 = REGIN
  ID2 = CNTN2 + MAXVEC
  COLOR2 = ID2 + MAXVEC
  ORIGINAL DRUM START ADDRESSES
  REGIN1 = COLOR2 + MAXVEC
  ID3 = ID2
  COLOR3 = COLOR2
  RETURN
  END

```

ORIGINAL PAGE IS
OF FOUR QUALITY

15-2

273

FILE: FLOCLS

```

SUBROUTINE FLOCLS(FIELDS,STAMNT,*,*,*,IPT,VERTEX)
FLOCLS GROUPS THE FIELDS CARDS ON A CLASS BASES FOR PROCESSING
IMPLICIT INTEGER (A-Z)
INCLUDE CMCHK1.LIST
INCLUDE CMCHK11.LIST
COMMON/INFORM/NOCLS2,NOISUR2,NOFFT2,VAHSZ2,TOTVT2,NOFLD2,
      AVAS2,COVAR2,CLS102,SURN02,SURNS2,FLOS2,VERTX2,
      FEVC2(30),SIHVC2(75),SIHPTW(75),CLSV2(60),
      KFPPTS(40),NOGRP,GRPNAM(60),GRPDIX(61),
      GRPCHK(41),GRHUPS(124)
COMMON/NDIM/NCLRCH,CLAVEC(30),MAXVEC,MAPKEY,
      CLASS,SIHCLS,FIELD,MEANSW,NOVEC,FODINF,*,SIZE,TOTMNS,
      CNTR1,CNTR2,IO1,IO2,COLOR1,COLOR2,HUFLEN,IO3,COLOR3,NODUMP,
      IDATA1,TOTVEC
CSEND
DIMENSION FIELDS(4,1),VERTEX(1)
LOGICAL SWITCH
DATA SWITCH/.TRUE./
IF (NOFLD2.EQ. 0) IPT = 1
IF (NOFLD2.EQ. 0) GO TO 75
IPT = IPT + FIELDS(4,NOFLD2)*2
75 CONTINUE
GO TO (40,100),STAMNT
80 1 = LAREAD(FIELDS(1,NOFLD2+1),VERTEX(IPT),FLOCLS(1),FIELDS(4,NOFLD2+1))
C
C WAS CLASS,SIHCLASS,FIELD, OR SEND* ENCOUNTERED
IF (1.EQ. -1) GO TO 90
IF (1.EQ. -2) GO TO 120
IF (1.EQ. 1) GO TO 130
IF (1.EQ. 0) GO TO 140
C
C CLASS CARD
90 IF (SWITCH) GO TO 100
STAMNT = 2
RETURN 2
100 READ(30,110)CLSV2(1)
DEVID = 30
IPT = 1
NOCLS2 = 1
NOFLD2 = 0
NOISUR2 = 0
SWITCH = .FALSE.
GO TO 80
C
C SURCLASS CARD
110 FORMAT(10X,A4)
120 NOISUR2 = NOISUR2 + 1
READ(30,110)SURVC2(NOISUR2)
DEVID = 30
GO TO 80
C
C FIELD CARD
130 NOFLD2 = NOFLD2 + 1
STAMNT = 1
RETURN 1
C
C SEND*
140 RETURN 3
C
END

```

FLO00010
 FLO00020
 FLO00030
 FLO00040
 FLO00050
 FLO00060
 FLO00070
 FLO00080
 FLO00090
 FLO00100
 FLO00110
 FLO00120
 FLO00130
 FLO00140
 FLO00150
 FLO00160
 FLO00170
 FLO00180
 FLO00190
 FLO00200
 FLO00210
 FLO00220
 FLO00230
 FLO00240
 FLO00250
 FLO00260
 FLO00270
 FLO00280
 FLO00290
 FLO00300
 FLO00310
 FLO00320
 FLO00330
 FLO00340
 FLO00350
 FLO00360
 FLO00370
 FLO00380
 FLO00390
 FLO00400
 FLO00410
 FLO00420
 FLO00430
 FLO00440
 FLO00450
 FLO00460
 FLO00470
 FLO00480
 FLO00490
 FLO00500
 FLO00510
 FLO00520
 FLO00530
 FLO00540
 FLO00550
 FLO00560
 FLO00570
 FLO00580
 FLO00590
 FLO00600
 FLO00610
 FLO00620
 FLO00630
 FLO00640
 FLO00650
 FLO00660
 FLO00670
 FLO00680

FILE: FLD0LD

```

SUBROUTINE FLD0LD(FIELDS,STAMNT,*,*,IPT,VERTEX)
C
C*
C* FLD0LD CONTROL THE PROCESSING OF FIELDS CARDS ON A PER FIELD BASES
C*
C* IMPLICIT INTEGER (A-Z)
C* INCLUDE COMM1.LIST
C* INCLUDE COMM11.LIST
C* COMMON/INFORM/NOCLS2,NOSUR2,NOFET2,VARSZ2,TOTVT2,NOFLD2,
C*   AVAR2,COVAR2,CLS1D2,SUMNO2,SUBDS2,FLOSV2,VERTX2,
C*   FETVC2(30),SUBVC2(75),SUPPTR(75),CLSVC2(60),
C*   KFPPTS(40),NOGRP,GRPNAM(60),GRPDEX(61),
C*   GRPCHK(61),GRGUPS(124)
C* COMMON /NDIM/NCLHCH,CLRVEC(30),MAXVEC,MAPKEY,
C*   CLASS,SURCLS,FIELD,MEANSW,NOVEC,FLDINF(6),SIZE,TOTMNS
C*   ,CNTP1,CNTR2,ID1,ID2,COLOR1,COLOR2,BUFLEN,ID3,COLOR3,NODUMP
C*   ,IDATA1,TUTVEC
CSEND
  DIMENSION FIELDS(4,1),VERTEX(1)
  NOFLD2 = 1
  IPT = 1
C
C
C  R0 I = LAREAD(FIELDS(1,NOFLD2),VERTEX(IPT),FLDINF(1),
C*   FIELDS(4,NOFLD2))
C
C  WAS CLASS,SURCLASS,FIELD, OR SEND* ENCOUNTERED
C
C  IF ( I .EQ. -1) GO TO 90
C  IF ( I .EQ. -2) GO TO 110
C  IF ( I .EQ. 1) GO TO 120
C  IF ( I .EQ. 0) GO TO 130
C
C  CLASS CARD
C
C  90 READ(30,100)CLSVC2(1)
C  REWIND 30
C  NOCLS2 = 1
C  100 FORMAT(10X,A4)
C  GO TO R0
C
C  SURCLASS CARD
C
C  110 READ(30,100)SURVC2(1)
C  REWIND 30
C  NOSUR2=1
C  GO TO R0
C
C  FIELD CARD
C
C  120 RETURN 1
C
C  SEND*
C
C  130 RETURN 2
C
C  END

```

FLD00010
 FLD00020
 FLD00030
 FLD00040
 FLD00050
 FLD00060
 FLD00070
 FLD00080
 FLD00090
 FLD00100
 FLD00110
 FLD00120
 FLD00130
 FLD00140
 FLD00150
 FLD00160
 FLD00170
 FLD00180
 FLD00190
 FLD00200
 FLD00210
 FLD00220
 FLD00230
 FLD00240
 FLD00250
 FLD00260
 FLD00270
 FLD00280
 FLD00290
 FLD00300
 FLD00310
 FLD00320
 FLD00330
 FLD00340
 FLD00350
 FLD00360
 FLD00370
 FLD00380
 FLD00390
 FLD00400
 FLD00410
 FLD00420
 FLD00430
 FLD00440
 FLD00450
 FLD00460
 FLD00470
 FLD00480
 FLD00490
 FLD00500
 FLD00510
 FLD00520
 FLD00530
 FLD00540
 FLD00550
 FLD00560
 FLD00570

FILE: FLDMEN

| | | |
|--------|---|----------|
| C* | SUBROUTINE FLDMEN(IDATA,J,NSAMP,NOFEAT,MEANS,RGCHAN,N) | FLD00010 |
| C* | FLDMEN COMPUTES THE FIELD MEANS | FLD00020 |
| C* | IMPLICIT INTEGER (A-Z) | FLD00030 |
| C | REAL MEANS,RND | FLD00040 |
| C | INCLUDE COMP1.LIST | FLD00050 |
| C | INCLUDE CMK11.LIST | FLD00060 |
| | COMMON/INFORM/NOCLS2,NOSUR2,NOFET2,VARS2,TOTVT2,NOFLD2, | FLD00070 |
| | AVAR2,COVAR2,CLS1D2,SURN02,SURDS2,FLDSV2,VERTX2, | FLD00080 |
| | FETVC2(30),SIBVC2(75),SURPTR(75),CLSV2(60), | FLD00090 |
| | KEPPTS(60),NOGRP,GRPNAM(60),GRPDEX(61), | FLD00100 |
| | GRPCHK(61),GROUPS(124) | FLD00110 |
| | COMMON /NDIM/NCLRCH,CLHVEC(30),MAXVEC,MAPKEY, | FLD00120 |
| | CLASS,SURCLS,FIELD,MEANSW,NOVEC,FLDINF(6),SIZE,TOTMNS | FLD00130 |
| | ,CNTR1,CNTR2,ID1,ID2,COLOR1,COLOR2,RUFLEN,ID3,COLOR3,NODUMP | FLD00140 |
| | ,IDATA1,TOTVEC | FLD00150 |
| C\$END | DIMENSION IDATA(NSAMP,NOFEAT),MEANS(NCLRCH,1) | FLD00160 |
| C | MEANS = ((N-1)*OLD MEAN)/N + DATA PT./N | FLD00170 |
| C | | FLD00180 |
| C | RND = (FLOAT(N)-1.0)/FLOAT(N) | FLD00190 |
| C | ICHAN = 0 | FLD00200 |
| C | DO 100 K = RGCHAN,NOFEAT | FLD00210 |
| | ICHAN = ICHAN + 1 | FLD00220 |
| 100 | MEANS(ICHAN,NOFLD2) = RND*MEANS(ICHAN,NOFLD2) + | FLD00230 |
| | FLOAT(IDATA(J,K))/FLOAT(N) | FLD00240 |
| C | RETURN | FLD00250 |
| | END | FLD00260 |
| | | FLD00270 |
| | | FLD00280 |
| | | FLD00290 |
| | | FLD00300 |
| | | FLD00310 |
| | | FLD00320 |
| | | FLD00330 |

~~15-6~~
277

ORIGINAL PAGE IS
OF POOR QUALITY

FILE: NDHST1

```

      RGCHAN = 1
      GO TO 135
102 CONTINUE
      DO 120 I=1,NCLRCH
120 FETVEC(NOFET2+I) = CLRVEC(I)
130 NOFEAT = NOFET2 + NCLRCH
C
      RGCHAN = NOFET2 + 1
135 CONTINUE
      COMPUTE ADDRESSES
C
      CALL ADDRES(TOP,NSAMP,NOFEAT,BEGIN,BEGIN)
C
      IF REMAINING STORAGE IN HIST ARRAY IS LESS THAN 2000, EMPTY
      HIST ARRAY (DATA VECTORS) ONTO DRUM, READ IN MAP TAPE,STORE ONTO
      DRUM, THEN READ DATA VECTORS BACK INTO HIST
C
      IF (MAPKEY.NE. 1) GO TO 105
      STORGE = LIMIT - NOVEC*SIZE
      IF (STORGE.LT. 2000) GO TO 103
      CALL STODAT(ILINE,NSAMP,HIST(NOVEC+1),STORGE,BEGIN)
      GO TO 105
103 VECTR1 = COLOR2
      WRDS = NOVEC*SIZE
      CALL RWRITE(VECTR1,HIST,WRDS,ISTAT)
104 IF (ISTAT.EQ. 1) GO TO 104
      REGIN1 = VECTR1 + WRDS
      CALL STODAT(ILINE,NSAMP,HIST,LIMIT,REGIN1)
      CALL RREAD(VECTR1,HIST,WRDS,ISTAT2)
C
      INITIALIZE IMAGE DATA TAPE
C
105 CALL TAPHDR(DATAPF,DATEFIL)
106 IF (ISTAT2.EQ. 1) GO TO 106
C
      POSITION IMAGE TAPE FOR THIS FIELD
      CALL FLDINT(FLDINF(1),FETVEC,NOFEAT)
      NLINE = 0
      NPTS = 0
C
      READ A SCAN LINE OF DATA AND PROCESS IT
C
      DO 500 LINE=LINSTR,LINEND,LININC
      NLINE = NLINE + 1
C
      CALL LINEHD(ARRAY(IDATA1),FNDTAP)
      IF (FNDTAP.EQ. -1) GO TO 600
C
      READ IN A SCAN LINE FROM CLASSIFICATION/CLUSTER MAP TAPE
      IF (MAPKEY.EQ. 1) CALL RFSTO(NLINE,NSAMP,BEGIN1)
C
      FIND INTERSECTIONS FOR N-P FIELDS
C
      CALL FDLINT(VERTEX(IPT),FIELDS(4,NOFLD2),FL,LINE,SAMP,NI)
C
      DO 400 J=1,NI,2
      IR = (FL(J)-SAMSTR)/SAMINC + 1
      IE = (FL(J+1)-SAMSTR)/SAMINC + 1
      IF (MOD(SAMSTR,SAMINC).NE. MOD(FL(J),SAMINC)) IB = IB + 1
      IF (IR.GT. IE) GO TO 400
C
      DO 350 K=IB,IE
C
      TOTVEC = TOTVEC + 1
      NPTS = NPTS + 1
      HISTOGRAM VECTOR
C
      CALL NDHST2(K,ARRAY(IDATA1),HIST,NOFET2,VECSWT,NSAMP,
      * ARRAY(1),VECCNT,OVRFLO,NOFEAT,RGCHAN)
C
      IF A NEW VECTOR WAS FOUND,VECSWT = 1
C
220 IF (MEANSW.EQ. 0) GO TO 330
      COMPUTE MEANS FOR TRAINING/TEST FIELDS

```

NDH00800
 NDH00810
 NDH00820
 NDH00830
 NDH00840
 NDH00850
 NDH00860
 NDH00870
 NDH00880
 NDH00890
 NDH00900
 NDH00910
 NDH00920
 NDH00930
 NDH00940
 NDH00950
 NDH00960
 NDH00970
 NDH00980
 NDH00990
 NDH01000
 NDH01010
 NDH01020
 NDH01030
 NDH01040
 NDH01050
 NDH01060
 NDH01070
 NDH01080
 NDH01090
 NDH01100
 NDH01110
 NDH01120
 NDH01130
 NDH01140
 NDH01150
 NDH01160
 NDH01170
 NDH01180
 NDH01190
 NDH01200
 NDH01210
 NDH01220
 NDH01230
 NDH01240
 NDH01250
 NDH01260
 NDH01270
 NDH01280
 NDH01290
 NDH01300
 NDH01310
 NDH01320
 NDH01330
 NDH01340
 NDH01350
 NDH01360
 NDH01370
 NDH01380
 NDH01390
 NDH01400
 NDH01410
 NDH01420
 NDH01430
 NDH01440
 NDH01450
 NDH01460
 NDH01470
 NDH01480
 NDH01490
 NDH01500
 NDH01510
 NDH01520
 NDH01530
 NDH01540
 NDH01550
 NDH01560
 NDH01570
 NDH01580

FILE: NDHST1

| | | |
|-----|--|----------|
| | CALL FLDPEN(ARRAY(IDATA),K,NSAMP,NOFEAT,MFANS,RCCHAN,NPTS) | NDH01590 |
| C | 330 IF (VECSWT .NE. 1) GO TO 350 | NDH01600 |
| C | ARRAY(ID1 + VECNT) = NOFLD2 | NDH01610 |
| | IF (MAPKEY .EQ. 0) GO TO 340 | NDH01620 |
| C | PETRIEVE CLUSTERED/CLASSIFIED DATA FROM DRUM | NDH01630 |
| C | CALL RSTOR(K,ARRAY(ID1+VECCNT)) | NDH01640 |
| C | 340 VECCNT = VECCNT + 1 | NDH01650 |
| | IF (VECCNT .LE. (RUFLN-1)) GO TO 350 | NDH01660 |
| C | DUMP ONTO DRUM | NDH01670 |
| C | NONDUMP = NONDUMP + 1 | NDH01680 |
| | CALL RWRITE(ID2,ARRAY(ID1),RUFLN,ISTAT1) | NDH01690 |
| | ID2 = ID2 + RUFLN | NDH01700 |
| | VECCNT = 0 | NDH01710 |
| | IF (MEANSW .EQ. 1) GO TO 350 | NDH01720 |
| | IF (INLRCH .EQ. 0) GO TO 350 | NDH01730 |
| | CALL RWRITE(COLOR2,ARRAY(COLOR1),RUFLN,ISTAT2) | NDH01740 |
| | COLOR2 = COLOR2 + RUFLN | NDH01750 |
| C | 350 CONTINUE | NDH01760 |
| C | 400 CONTINUE | NDH01770 |
| C | 500 CONTINUE | NDH01780 |
| C | IF OVRFLD IS GREATER THAN 0, HISTOGRAMMED DATA VECTOR TABLE IS FULL | NDH01790 |
| C | IF (OVRFLD .EQ. 0) GO TO 505 | NDH01800 |
| | WRITE(6,221) | NDH01810 |
| 221 | FORMAT(////) | NDH01820 |
| | WRITE(6,225)OVRFLD | NDH01830 |
| 225 | FORMAT(1X,16,' VECTORS WERE NOT HISTOGRAMMED, BUT USED IN COMPUTING MEANS, IF APPLICABLE') | NDH01840 |
| 505 | CONTINUE | NDH01850 |
| | IF (FIELD .NE. 1) GO TO 65 | NDH01860 |
| | CALL RRTFIL(HIST,MEANS,ARRAY(ID1),ARRAY(COLOR1),FIELDS,VERTEX, | NDH01870 |
| | * TENTP) | NDH01880 |
| | GO TO 90 | NDH01890 |
| C | WRITE HISTOGRAM FILE | NDH01900 |
| C | 510 CALL RRTFIL(HIST,MEANS,ARRAY(ID1),ARRAY(COLOR1),FIELDS,VERTEX, | NDH01910 |
| | * TENTP) | NDH01920 |
| | GO TO 90 | NDH01930 |
| C | SEND CARD FOUND | NDH01940 |
| C | 520 CALL RRTFIL(HIST,MEANS,ARRAY(ID1),ARRAY(COLOR1),FIELDS,VERTEX, | NDH01950 |
| | * TENTP) | NDH01960 |
| 530 | ENDFILE NHSTUN | NDH01970 |
| | DEFIND NHSTUN | NDH01980 |
| | RETURN | NDH01990 |
| 600 | WRITE(6,610) | NDH02000 |
| 610 | FORMAT(' ERROR IN FIELD CARD. ABORTING') | NDH02010 |
| | CALL CWERH | NDH02020 |
| | END | NDH02030 |
| | | NDH02040 |
| | | NDH02050 |
| | | NDH02060 |
| | | NDH02070 |
| | | NDH02080 |
| | | NDH02090 |
| | | NDH02100 |
| | | NDH02110 |
| | | NDH02120 |
| | | NDH02130 |
| | | NDH02140 |
| | | NDH02150 |
| | | NDH02160 |
| | | NDH02170 |
| | | NDH02180 |
| | | NDH02190 |
| | | NDH02200 |
| | | NDH02210 |

ALL PAGE IS
OF POOR QUALITY

FILE: NDHST2

```

      SURROUTINE NDHST2(J, IDATA, HIST, NOFET2, VEC SWT, NSAMP, ARRAY, VEC CNT,
      * OVFFLO, NOFEAT, RGCHAN)
C
C* NDHST2 PERFORMS THE 1 TO 16 CHANNEL HISTOGRAM
C* THE HISTOGRAM IS COMPUTED FOR EITHER ONE OR TWO SETS OF CHANNELS
C
      IMPLICIT INTEGER (A-Z)
      DIMENSION HIST(SIZE, MAXVEC), IDATA(1), COMWRD(4)
      DIMENSION ARRAY(1)
      INCLUDE COMMK6.LIST
C
      INCLUDE COMMK11.LIST
      COMMON/GLOBAL/HEAD(63), MAPTAP, DATAPE, SAVTAP, BMFILE, BMKEY,
      * HISFIL, HISKEY, TRFORM, EKIPTP, ERPKEY, MAPUNT, NOFILF,
      * DRIMAD, DRMWDS, PAGESIZ, DATFIL, STAFIL, ASAV, ASAVFL
      * NHSTUN, NHSTFI, SCTRUN, MAPFIL
      * DOTUNT, DOTFIL, NCHPAS, TRNSFL, BMTRFL, HISTFL, PCHUNT,
      * CPDUNT, PRUNT, RANDIO
      COMMON /NDIM/NCLCH, CLRVEC(30), MAXVEC, MAPKEY,
      * CLASS, SURCLS, FIELD, MEANS*, NOVEC, FLDINF(6), SIZE, TOTMNS
      * CNTR1, CNTR2, ID1, ID2, COLOR1, COLOR2, BUFLN, ID3, COLOR3, NONUMP
      * IDATA1, TOTVEC
CSEND
C
      VEC SWT = 0
C
      LOGICAL*1 LDUM(4), LLDUM(4)
      EQUIVALENCE (IDUM, LDUM(1)), (IIDUM, LLDUM(1))
      DO 50 I=1, 4
      50 COMWRD(I)=0
      II=0
      DO 60 I=1, NOFET2
      II=II+1
      III=1+(II-1)/4
      IIII=J+(II-1)*NSAMP
      IYTE=II-(II-1)/4
      IF (IYTE.EQ.1) IIDUM=0
      IDUM=IDATA(IIII)
      LDUM(IYTE)=LDUM(4)
      COMWRD(IIII)=IIDUM
      60 CONTINUE
C
      STORE LAST 8 BITS OF IDATA(IIII) INTO COMWRD(IIII) IN
      C RYTE POSITION IYTE, LEFT TO RIGHT
      IF (NCLCH.EQ.0) GO TO 100
      IF (MEANS*.EQ.1) GO TO 100
      CALL PICOLR(IDATA, J, NOFEAT, COLWRD, NSAMP, NOFET2, RGCHAN)
      100 CONTINUE
      IF (NOVEC.EQ.0) GO TO 135
C
      COMPARING VECTORS ALREADY FOUND WITH IN COMING VECTOR
C
      110 DO 130 K=1, NOVEC
C
      DO 120 L=1, SIZE
      IF (COMWRD(L).NE. HIST(L, K)) GO TO 130
      120 CONTINUE
C
      FOUND A MATCHING VECTOR
C
      DO ONE OF THE FOLLOWING :
      1) HISTOGRAM THE VECTOR ONLY IF COLOR CODES FOR BOTH VECTORS ARE
      THE SAME.
      2. DO NOT CHECK COLOR CODES IF MAP TAPE IS BEING INPUT OR MEANS
      FOR FIELDS ARE BEING COMPUTED
C
      IF (NCLCH.EQ.0) GO TO 126
      IF (MEANS*.EQ.1) GO TO 126
C
      IS COLOR CODE IN CORE OR ON DRUM
C
      IF (NONUMP.EQ.0) GO TO 122
      IF (K.GT. (NONUMP*BUFLN)) GO TO 122
      ADDRES = COLOR3 + (NONUMP-1)*BUFLN + K - 1
      CALL HPEAD(ADDRES, COFS, 1, ISTAT2)
      121 IF (ISTAT2.EQ.1) GO TO 121
      GO TO 124
      122 KK = K - (NONUMP*BUFLN)
      COFS = A*RAY(COLOR1 + KK - 1)
      124 IF (COFS.NE. COLWRD) GO TO 150

```

NDH00010
 NDH00020
 NDH00030
 NDH00040
 NDH00050
 NDH00060
 NDH00070
 NDH00080
 NDH00090
 NDH00100
 NDH00110
 NDH00120
 NDH00130
 NDH00140
 NDH00150
 NDH00160
 NDH00170
 NDH00180
 NDH00190
 NDH00200
 NDH00210
 NDH00220
 NDH00230
 NDH00240
 NDH00250
 NDH00260
 NDH00270
 NDH00280
 NDH00290
 NDH00300
 NDH00310
 NDH00320
 NDH00330
 NDH00340
 NDH00350
 NDH00360
 NDH00370
 NDH00380
 NDH00390
 NDH00400
 NDH00410
 NDH00420
 NDH00430
 NDH00440
 NDH00450
 NDH00460
 NDH00470
 NDH00480
 NDH00490
 NDH00500
 NDH00510
 NDH00520
 NDH00530
 NDH00540
 NDH00550
 NDH00560
 NDH00570
 NDH00580
 NDH00590
 NDH00600
 NDH00610
 NDH00620
 NDH00630
 NDH00640
 NDH00650
 NDH00660
 NDH00670
 NDH00680
 NDH00690
 NDH00700
 NDH00710
 NDH00720
 NDH00730
 NDH00740
 NDH00750
 NDH00760
 NDH00770
 NDH00780
 NDH00790

FILE: NDHST2

| | | | |
|---|-----|--|----------|
| C | 126 | CONTINUE | NDH00800 |
| | | CNTR1 = CNTR2 + K - 1 | NDH00810 |
| | | CALL RPFAD(CNTR1,COUNTR,1,ISTAT) | NDH00820 |
| | 125 | IF(ISTAT.EQ.1) GO TO 125 | NDH00830 |
| | | COUNTR = COUNTR + 1 | NDH00840 |
| | | CALL RWRITE(CNTR1,COUNTR,1,ISTAT1) | NDH00850 |
| | 127 | IF(ISTAT1.EQ.1) GO TO 127 | NDH00860 |
| | | GO TO 150 | NDH00870 |
| | 130 | CONTINUE | NDH00880 |
| C | | | NDH00890 |
| C | | INSERT NEW VECTOR | NDH00900 |
| C | | | NDH00910 |
| | 135 | IF (NOVEC .LT. MAXVEC) GO TO 137 | NDH00920 |
| | | OVRFLO = OVRFLO + 1 | NDH00930 |
| | | IF (OVRFLO .GT. 1) GO TO 136 | NDH00940 |
| | | WRITE(6,133) | NDH00950 |
| | 133 | FORMAT(////////) | NDH00960 |
| | | WRITE(4,123)MAXVEC | NDH00970 |
| | 123 | FORMAT(/' CORE LIMITS EXCEEDED. MAXIMUM NO. OF VECTORS'/' ACCEPTED | NDH00980 |
| | | * IS ',I6) | NDH00990 |
| C | | | NDH01000 |
| C | | VECTOR TABLE IS FULL--CONTINUE TO HISTOGRAM DATA VECTORS THAT | NDH01010 |
| C | | ALREADY EXIST | NDH01020 |
| C | | | NDH01030 |
| | 136 | RETURN | NDH01040 |
| | 137 | NOVEC = NOVEC + 1 | NDH01050 |
| | | DO 140 L=1,SIZE | NDH01060 |
| | 140 | HIST(L,NOVEC) = COMWRD(L) | NDH01070 |
| | | CNTR1 = CNTR2 + NOVEC - 1 | NDH01080 |
| | | COUNTR = 1 | NDH01090 |
| | | CALL RWRITE(CNTR1,COUNTR,1,ISTAT3) | NDH01100 |
| | 145 | IF (ISTAT3.EQ.1) GO TO 145 | NDH01110 |
| | | VECSWT = 1 | NDH01120 |
| | | IF (MFANSW.NE.0.OR. MAPKEY.NE.0) GO TO 150 | NDH01130 |
| | | ARRAY(COLOR1 + VEC CNT) = COLWRD | NDH01140 |
| | 150 | CONTINUE | NDH01150 |
| | 200 | CONTINUE | NDH01160 |
| C | | | NDH01170 |
| | | RETURN | NDH01180 |
| | | END | NDH01190 |
| | | | NDH01200 |

FILE: PICOLR

| | | |
|-------|--|----------|
| C* | SUBROUTINE PICOLR(IDATA,K,NOFEAT,COLWRD,NSAMP,NOFET2,RGCHAN) | PIC00010 |
| C* | PICOLR EXTRACT THE COLOR CHANNELS FROM IDATA AND | PIC00020 |
| C* | PACK THE 8 BIT PIXELS INTO A COMPUTER WORD | PIC00030 |
| C | IMPLICIT INTEGER (A-Z) | PIC00040 |
| | INCLUDE CMH11.LIST | PIC00050 |
| | COMMON /NDIM/NCLRCH,CLRVEC(30),MAXVEC,MAPKEY, | PIC00060 |
| | CLASS,SUBCLS,FIELD,MFANSW,NOVEC,FLDINF(6),SIZE,TOTMNS | PIC00070 |
| | ,CNTB1,CNTR2,ID1,ID2,COLOR1,COLOR2,HUFLEN,ID3,COLOR3,NODIMP | PIC00080 |
| | ,IDATA1,TOTVEC | PIC00090 |
| CSEND | | PIC00100 |
| | DIMENSION IDATA(1) | PIC00110 |
| | LOGICAL*1 LDUM(4),LLDUM(4) | PIC00120 |
| | EQUIVALENCE (IDUM,LDUM(1)),(IIDUM,LLDUM(1)) | PIC00130 |
| | IIDUM=0 | PIC00140 |
| | DO 100 I=1,NCLRCH | PIC00150 |
| | II=K+(RGCHAN+I-2)*NSAMP | PIC00160 |
| | IDUM=IDATA(II) | PIC00170 |
| 100 | LLDUM(I)=LDUM(4) | PIC00180 |
| | COLWRD=IIDUM | PIC00190 |
| | RETURN | PIC00200 |
| C | | PIC00210 |
| | END | PIC00220 |
| | | PIC00230 |

FILE: RESTO

| | | |
|-----|--|----------|
| C | SUBROUTINE RESTO (ILINE,NSAMP,REGIN1) | RES00010 |
| C | | RES00020 |
| C | RESTO RETRIEVES THE CLUSTER MAP FROM DRUM A LINE AT A TIME | RES00030 |
| C | RESTOR PICKS OUT THE DESIRED ID FROM WITHIN THE LINE | RES00040 |
| C | | RES00050 |
| C | IMPLICIT INTEGER (A-Z) | RES00060 |
| C | INCLUDE COMPAK.LIST | RES00070 |
| C | COMMON/GLOBAL/HEAD(63),MARTAP,DATAPF,SAVTAP,BMFILE,BMKEY, | RES00080 |
| | HISFIL,HISKFY,INFORM,ERIPTP,EPKEY,MAPUNT,NOFILF, | RES00090 |
| | DRUMAD,DCMADS,PAGSIZ,DATFIL,STAFIL,ASAV,ASAVFL | COM00010 |
| | .NHSTUN,NHSTFI,SCTRUN,MAFFIL | COM00020 |
| | .DOTUNT,DOTFIL,NCHPAS,TRNSFL,BMTRFL,HISTFL,PCHUNT, | COM00030 |
| | CRDUNT,ERTUNT,RANDIO | COM00040 |
| C | COMMON /IDWORD/ IDWORD(1000) | COM00050 |
| C | ADDRES = REGIN1 + (ILINE-1)*NSAMP | COM00060 |
| C | CALL RREAD(ADDRES,IDWORD,NSAMP,ISTAT) | RES00110 |
| 150 | IF (ISTAT.EQ. 1) GO TO 150 | RES00120 |
| | IF (ISTAT.GE. 0) RETURN | RES00130 |
| | WRITE(6,100) | RES00140 |
| 100 | FORMAT(' EPHOR READING DRUM') | RES00150 |
| | CALL CWERP | RES00160 |
| C | | RES00170 |
| C | ENTRY RESTOR(K,NUMB) | RES00180 |
| C | | RES00190 |
| C | NUMR = IDWORD(K) | RES00200 |
| C | RETURN | RES00210 |
| C | END | RES00220 |
| | | RES00230 |
| | | RES00240 |
| | | RES00250 |
| | | RES00260 |
| | | RES00270 |
| | | RES00280 |
| | | RES00290 |
| | | RES00300 |

ORIGINAL PAGE IS
OF POOR QUALITY

SET00010
SET00020
SET00030
SET00040
SET00050
SET00060
SET00070
SET00080
SET00090
SET00100
SET00110
SET00120
SET00130
SET00140
SET00150
SET00160
SET00170
SET00180
SET00190
SET00200
SET00210
SET00220
SET00230
SET00240
SET00250
SET00260
SET00270
SET00280
SET00290
SET00300
SET00310
SET00320
SET00330
SET00340
SET00350
SET00360
SET00370
SET00380
SET00390
SET00400
SET00410
SET00420
SET00430
SET00440
SET00450
SET00460
SET00470
SET00480
SET00490
SET00500
SET00510
SET00520
SET00530
SET00540
SET00550
SET00560
SET00570
SET00580
SET00590
SET00600
SET00610
SET00620
SET00630
SET00640
SET00650
SET00660
SET00670
SET00680
SET00690
SET00700
SET00710
SET00720
SET00730
SET00740
SET00750
SET00760
SET00770
SET00780
SET00790

FILE: SET10

C CHANNEL CARD

```

150 M = NKTCHW(CARD,COL)
IF (M.EQ. BLANK) GO TO 105
IF (M.EQ. PHCD) GO TO 150
IF (M.EQ. CHCD) GO TO 170
153 WRITE(A,155)
155 FORMAT(' ERROR ON CHANNELS CARD')
GO TO 105
160 J = FIND12(CARD,COL,EQUOM)
IF (J.NE. 2) GO TO 153
NOFET2 = NUMBER(CARD,COL,FETVC2,NOFET2)
COL = COL - 1
CALL ORDER(FETVC2,NOFET2)
GO TO 150
170 J = FIND12(CARD,COL,EQUOM)
IF (J.NE. 2) GO TO 153
NCLRCH = NUMBER(CARD,COL,CLRVEC,NCLRCH)
COL = COL - 1
CALL ORDER(CLRVEC,NCLRCH)
GO TO 150

```

C DATA FILE CARD

```

180 M = NKTCHW(CARD,COL)
IF (M.EQ. BLANK) GO TO 105
IF (M.EQ. UNCD) GO TO 190
IF (M.EQ. FPCD) GO TO 200
185 WRITE(A,187)
187 FORMAT(' ERROR ON DATA FILE CARD')
GO TO 105
190 J = FIND12(CARD,COL,EQUOM)
IF (J.NE. 2) GO TO 185
M = NUMBER(CARD,COL,DATAPE,ZERO)
COL = COL - 1
GO TO 180
200 J = FIND12(CARD,COL,EQUOM)
IF (J.NE. 2) GO TO 185
M = NUMBER(CARD,COL,DATAFIL,ZERO)
DATAFIL = DATAFIL - 1
COL = COL - 1
GO TO 180

```

C CLUSTER/CLASSIFICATION MAP TAPE

```

210 M = NKTCHW(CARD,COL)
IF (M.EQ. BLANK) GO TO 105
IF (M.EQ. UNCD) GO TO 230
IF (M.EQ. FPCD) GO TO 240
215 WRITE(A,220)
220 FORMAT(' ERROR ON UAS FILE CARD')
GO TO 105
230 J = FIND12(CARD,COL,EQUOM)
IF (J.NE. 2) GO TO 215
M = NUMBER(CARD,COL,MAPUNT,ZERO)
MAPKEY = 1
COL = COL - 1
GO TO 210
240 J = FIND12(CARD,COL,EQUOM)
IF (J.NE. 2) GO TO 215
M = NUMBER(CARD,COL,MAPFIL,ZERO)
MAPFIL = MAPFIL - 1
COL = COL - 1
GO TO 210

```

C N-DIM HISTOGRAM FILE

```

250 M = NKTCHW(CARD,COL)
IF (M.EQ. BLANK) GO TO 105
IF (M.EQ. UNCD) GO TO 270
IF (M.EQ. FPCD) GO TO 280
260 WRITE(A,265)
265 FORMAT(' ERROR ON N-DIM HISTOGRAM FILE CARD')
GO TO 105
270 J = FIND12(CARD,COL,EQUOM)
IF (J.NE. 2) GO TO 260
M = NUMBER(CARD,COL,NHSTUN,ZERO)
COL = COL - 1
GO TO 250

```

SET 100800
SET 100810
SET 100820
SET 100830
SET 100840
SET 100850
SET 100860
SET 100870
SET 100880
SET 100890
SET 100900
SET 100910
SET 100920
SET 100930
SET 100940
SET 100950
SET 100960
SET 100970
SET 100980
SET 100990
SET 101000
SET 101010
SET 101020
SET 101030
SET 101040
SET 101050
SET 101060
SET 101070
SET 101080
SET 101090
SET 101100
SET 101110
SET 101120
SET 101130
SET 101140
SET 101150
SET 101160
SET 101170
SET 101180
SET 101190
SET 101200
SET 101210
SET 101220
SET 101230
SET 101240
SET 101250
SET 101260
SET 101270
SET 101280
SET 101290
SET 101300
SET 101310
SET 101320
SET 101330
SET 101340
SET 101350
SET 101360
SET 101370
SET 101380
SET 101390
SET 101400
SET 101410
SET 101420
SET 101430
SET 101440
SET 101450
SET 101460
SET 101470
SET 101480
SET 101490
SET 101500
SET 101510
SET 101520
SET 101530
SET 101540
SET 101550
SET 101560
SET 101570
SET 101580

ORIGINAL PAGE IS
OF POOR QUALITY

| | | | | | | |
|---|---|---|---|---|---|---|
| S | T | 0 | 1 | 5 | 9 | 0 |
| S | T | 0 | 1 | 6 | 0 | 0 |
| S | T | 0 | 1 | 6 | 1 | 0 |
| S | T | 0 | 1 | 6 | 2 | 0 |
| S | T | 0 | 1 | 6 | 3 | 0 |
| S | T | 0 | 1 | 6 | 4 | 0 |
| S | T | 0 | 1 | 6 | 5 | 0 |
| S | T | 0 | 1 | 6 | 6 | 0 |
| S | T | 0 | 1 | 6 | 7 | 0 |
| S | T | 0 | 1 | 6 | 8 | 0 |
| S | T | 0 | 1 | 6 | 9 | 0 |
| S | T | 0 | 1 | 7 | 0 | 0 |
| S | T | 0 | 1 | 7 | 1 | 0 |
| S | T | 0 | 1 | 7 | 2 | 0 |
| S | T | 0 | 1 | 7 | 3 | 0 |
| S | T | 0 | 1 | 7 | 4 | 0 |
| S | T | 0 | 1 | 7 | 5 | 0 |
| S | T | 0 | 1 | 7 | 6 | 0 |
| S | T | 0 | 1 | 7 | 7 | 0 |
| S | T | 0 | 1 | 7 | 8 | 0 |
| S | T | 0 | 1 | 7 | 9 | 0 |
| S | T | 0 | 1 | 8 | 0 | 0 |
| S | T | 0 | 1 | 8 | 1 | 0 |
| S | T | 0 | 1 | 8 | 2 | 0 |
| S | T | 0 | 1 | 8 | 3 | 0 |
| S | T | 0 | 1 | 8 | 4 | 0 |
| S | T | 0 | 1 | 8 | 5 | 0 |
| S | T | 0 | 1 | 8 | 6 | 0 |
| S | T | 0 | 1 | 8 | 7 | 0 |
| S | T | 0 | 1 | 8 | 8 | 0 |
| S | T | 0 | 1 | 8 | 9 | 0 |
| S | T | 0 | 1 | 9 | 0 | 0 |
| S | T | 0 | 1 | 9 | 1 | 0 |
| S | T | 0 | 1 | 9 | 2 | 0 |
| S | T | 0 | 1 | 9 | 3 | 0 |
| S | T | 0 | 1 | 9 | 4 | 0 |
| S | T | 0 | 1 | 9 | 5 | 0 |
| S | T | 0 | 1 | 9 | 6 | 0 |
| S | T | 0 | 1 | 9 | 7 | 0 |
| S | T | 0 | 1 | 9 | 8 | 0 |
| S | T | 0 | 1 | 9 | 9 | 0 |
| S | T | 0 | 2 | 0 | 0 | 0 |
| S | T | 0 | 2 | 0 | 1 | 0 |
| S | T | 0 | 2 | 0 | 2 | 0 |
| S | T | 0 | 2 | 0 | 3 | 0 |
| S | T | 0 | 2 | 0 | 4 | 0 |
| S | T | 0 | 2 | 0 | 5 | 0 |
| S | T | 0 | 2 | 0 | 6 | 0 |
| S | T | 0 | 2 | 0 | 7 | 0 |
| S | T | 0 | 2 | 0 | 8 | 0 |
| S | T | 0 | 2 | 0 | 9 | 0 |
| S | T | 0 | 2 | 1 | 0 | 0 |
| S | T | 0 | 2 | 1 | 1 | 0 |
| S | T | 0 | 2 | 1 | 2 | 0 |
| S | T | 0 | 2 | 1 | 3 | 0 |
| S | T | 0 | 2 | 1 | 4 | 0 |
| S | T | 0 | 2 | 1 | 5 | 0 |
| S | T | 0 | 2 | 1 | 6 | 0 |
| S | T | 0 | 2 | 1 | 7 | 0 |
| S | T | 0 | 2 | 1 | 8 | 0 |
| S | T | 0 | 2 | 1 | 9 | 0 |
| S | T | 0 | 2 | 2 | 0 | 0 |
| S | T | 0 | 2 | 2 | 1 | 0 |
| S | T | 0 | 2 | 2 | 2 | 0 |
| S | T | 0 | 2 | 2 | 3 | 0 |
| S | T | 0 | 2 | 2 | 4 | 0 |
| S | T | 0 | 2 | 2 | 5 | 0 |
| S | T | 0 | 2 | 2 | 6 | 0 |
| S | T | 0 | 2 | 2 | 7 | 0 |
| S | T | 0 | 2 | 2 | 8 | 0 |
| S | T | 0 | 2 | 2 | 9 | 0 |
| S | T | 0 | 2 | 3 | 0 | 0 |
| S | T | 0 | 2 | 3 | 1 | 0 |
| S | T | 0 | 2 | 3 | 2 | 0 |
| S | T | 0 | 2 | 3 | 3 | 0 |
| S | T | 0 | 2 | 3 | 4 | 0 |
| S | T | 0 | 2 | 3 | 5 | 0 |
| S | T | 0 | 2 | 3 | 6 | 0 |
| S | T | 0 | 2 | 3 | 7 | 0 |

FILE: SET10

```
1010 FORMAT(' HISTOGRAM DATA VECTORS FROM CHANNELS ',16(I2.1X))
1020 FORMAT(' COLOR CODES ARE FROM CHANNELS ',4(I2.1X))
1030 FORMAT(' CLUSTER/CLASSIFICATION TAPE IS BEING INPUT')
1040 FORMAT(' HISTOGRAM FIELDS BY CLASS')
1050 FORMAT(' HISTOGRAM FIELDS BY SUBCLASS')
1060 FORMAT(' HISTOGRAM FIELDS ON PEN FIELDS BASES')
1070 FORMAT(' COMPUTE MEANS OF INPUT FIELDS')
      RETURN
```

C

END

SET02380
SET02390
SET02400
SET02410
SET02420
SET02430
SET02440
SET02450
SET02460
SET02470

ORIGINAL PAGE 1
OF POOR QUALITY

FILE: STODAT

```

C* SURROUTINE STODAT(ILINE,NSAMP,HIST,LIMIT,BEGIN1)
C* STODAT READS AND STORES THE CLASSIFICATION/CLUSTER MAP ON DRUM
C* IMPLICIT INTEGER (A-Z)
C INCLUDE COMRAK6.LIST
COMMON/GLOBAL/HEAD(63),MAPTAP,DATAP,SAVTAP,BMFILE,BMKEY,
* HISFIL,HISKEY,TRFORM,ERPTP,ERPKEY,MAPUNT,NOFILF,
* DRUMAD,DRMWD,DRMWD,DRMWD,PAGSIZ,DATFIL,STAFIL,ASAV,ASAVFL
* ,NHSTUN,NHSTFI,SCTRUN,MAPFIL
* ,DOTUNT,DOTFIL,NCHPAS,TRNSFL,BMTRFL,HISTFL,PCHUNT,
* CRDUNT,PRUNT,RANDIO
CSEND
DIMENSION HIST(LIMIT),FETVEC(1),FLD(6),NLINE(4)
TOTWRD = ILINE*NSAMP
IF (TOTWRD .LE. (DRMWD-(DRUMAD-BEGIN1))) GO TO 120
WRITE(6,110)
110 FORMAT(' NOT ENOUGH DRUM SPACE TO STORE DAS TAPE DATA',
CALL CMERR
C 120 CALL TAPHOR(MAPUNT,MAPFIL)
FETVEC(1) = 1
NOFEAT = 1
FLD(1) = 1
FLD(2) = ILINE
FLD(3) = 1
FLD(4) = 1
FLD(5) = NSAMP
FLD(6) = 1
REGIN = BEGIN1
CALL FLOINT(FLD ,FETVEC,NOFEAT)
DUMPS = TOTWRD / LIMIT
IF (MOD(TOTWRD,LIMIT) .NE. 0) DUMPS = DUMPS + 1
TOTLNS = LIMIT / NSAMP
IF (TOTLNS .GE. ILINE) GO TO 140
DMP = DUMPS - 1
DO 130 I=1,DMP
130 NLINE(I) = TOTLNS
NLINE(DUMPS) = ILINE - TOTLNS*DMP
GO TO 150
140 NLINE(1) = ILINE
C 150 DO 200 J=1,DUMPS
NUMLIN = NLINE(J)
DO 160 K=1,NUMLIN
WORDS = NSAMP*(K-1)
160 CALL LINERD(HIST(WORDS+1),ENDTAP)
C
C STORE ON HIGH SPEED DRUM
C
NWORDS = WORDS + NSAMP
CALL RWRITE(REGIN,HIST(1),NWORDS,ISTAT)
200 REGIN = REGIN + NLINE(J) * NSAMP
C
MAPFIL = MAPFIL + 1
C
RETURN
END

```

ST000010
ST000020
ST000030
ST000040
ST000050
ST000060
COM00010
COM00020
COM00030
COM00040
COM00050
COM00060
ST000080
ST000090
ST000100
ST000110
ST000120
ST000130
ST000140
ST000150
ST000160
ST000170
ST000180
ST000190
ST000200
ST000210
ST000220
ST000230
ST000240
ST000250
ST000260
ST000270
ST000280
ST000290
ST000300
ST000310
ST000320
ST000330
ST000340
ST000350
ST000360
ST000370
ST000380
ST000390
ST000400
ST000410
ST000420
ST000430
ST000440
ST000450
ST000460
ST000470
ST000480
ST000490
ST000500
ST000510
ST000520
ST000530

```

SUBROUTINE WRTFIL(HIST,MEANS,ID,COLOR,FIELDS,VERTEX,I)
WRTFIL WRITES THE NDIM FILE
IMPLICIT INTEGER (A-Z)
INCLUDE COMAK1.LIST
INCLUDE COMAK6.LIST
INCLUDE CMK11.LIST
COMMON/INFORM/NOSUR2,NOFET2,VARSZ2,TOTVT2,NOFLD2,
* AVAP2,CQVAR2,CLSID2,SURN02,SURDS2,FLDSV2,VERTX2,
* FETVC2(30),SURVC2(75),SURPTR(75),CLSVC2(60),
* KEPPTS(60),NOGRP,GRPNAM(60),GRPDEX(61),
* GRPCHK(61),GROUPS(124)
COMMON/GLOBAL/HEAD(63),MAPTAP,DATEP,SAVTAP,RMFILE,RMKEY,
* HISFIL,HISKEY,THFORM,ERIPTR,ERPKEY,MAPUNT,NOFILE,
* DRUMAD,DRM4DS,PAGSIZ,DATFIL,STAFIL,ASAV,ASAVFL
* ,NHSTUN,NHSTFI,SCTRIIN,MAPFIL
* ,DOTUNT,DOTFIL,NCHPAS,TRNSFL,BMTWFL,HISTFL,PCHUNT,
* CRDUNT,PHTUNT,RANDIO
COMMON/NDIM/NCLRCH,CLRVEC(30),MAXVEC,MAPKEY,
* CLASS,SUBCLS,FIELD,MEANSW,NOVEC,FLDINF(6),SIZE,TOTMNS
* ,CNTR1,CNTR2,ID1,ID2,COLOR1,COLOR2,RUFLEN,ID3,COLOR3,NODUMP
* ,IDATA1,TOTVEC
CSEND
DIMENSION HIST(1),MEANS(1),ISTAT(4),ID(1),COLOR(1),VERTEX(2,1)
DIMENSION FIELDS(4,1)
REAL MEANS
DATA BLANK/' '/
I = I + 1
IF (I .GT. 1) GO TO 100
WRITE HEADER RECORD
WRITE(NHSTUN) TOTMNS,SIZE,NOFET2,(FETVC2(I),I=1,NOFET2),
* NCLRCH,(CLRVEC(I),I=1,NCLRCH)
ENDFILE NHSTUN
WRITE RECORD 1
100 WRITE(NHSTUN) NOFLD2,NOSUR2,TOTVT2,NOVEC
WRITE RECORD 2
WRITE(NHSTUN) CLSVC2(1),(SURVC2(I),I=1,NOSUR2),((FIELDS(I,J),
* I=1,4),J=1,NOFLD2),((VERTEX(I,J),I=1,2),J=1,TOTVT2)
WRITE RECORD 3
IF (TOTMNS .GT. 0) WRITE(NHSTUN) (MEANS(I),I=1,TOTMNS)
WRITE RECORD 4
II = SIZE*NOVEC
WRITE(NHSTUN) (HIST(I),I=1,II)
IF (NODUMP .EQ. 0) GO TO 160
PEAD ID INFORMATION INTO CORE
WRITE RECORD 5
VECDRM = NODUMP*RUFLEN
VECCARY = NOVEC - VECDRM
CALL RREAD(ID3,HIST(1),VECDRM,ISTAT(1))
IF (VECCARY .EQ. 0) GO TO 115
DO 110 I=1,VECCARY
110 HIST(VECDRM+I) = ID(I)
115 CONTINUE
120 IF (ISTAT(1) .EQ. 1) GO TO 125
WRITE(NHSTUN) (HIST(I),I=1,NOVEC)
WRITE REC 6
CALL RREAD(CNTR2,HIST(1),NOVEC,ISTAT(3))
125 IF (ISTAT(3) .EQ. 1) GO TO 125
WRITE(NHSTUN) (HIST(I),I=1,NOVEC)
WRITE REC 7

```

FILE: WRTFIL

```

      IF (TOTMNS .NE. 0) GO TO 180
      IF (NCLRCH .EQ. 0) GO TO 180
      CALL RREAD(COLW3,HIST(1),VECDRM,ISTAT(2))
      IF (VECARV .EQ. 0) GO TO 135
      DO 130 I=1,VECARV
130  HIST(VECDRM+I) = COLOR(I)
135  IF (ISTAT(2) .EQ. 1) GO TO 135
140  WRITE(NHSTUN)(HIST(I),I=1,NOVEC)
      GO TO 180

      INFORMATION DID NOT NEED TO BE STORED ON DRUM

      RECORD 5
160  WRITE(NHSTUN)(ID(I),I=1,NOVEC)
      RECORD 6

      HISTOGRAM INFORMATION IS ALWAYS STORED ON HIGH SPEED FROM
165  CALL RREAD(CNTR2,HIST(1),NOVEC,ISTAT(3))
170  IF (ISTAT(3) .EQ. 1) GO TO 170
      WRITE(NHSTUN)(HIST(I),I=1,NOVEC)

      WRITE REC 7

      IF (TOTMNS .NE. 0) GO TO 180
      IF (NCLRCH .GT. 0) WRITE(NHSTUN)(COLOR(I),I=1,NOVEC)
180  ENDFILE NHSTUN

      WRITE FILE INFO ON LINE PRINTER
      CALL WRTFLD(FIELDS,VERTEX,NOFLD2,2,CLSV2,SURVC2)
      WRITE(6,190)TOTVEC,NOVEC
190  FORMAT(///55X,'TOTAL NO. OF VECTORS =',I6/48X,'TOTAL NO. OF UNIO
      *UE VECTORS =',I6)

      IF (TOTMNS .EQ. 0) RETURN
      WRITE(6,200)(BLANK,CLRVEC(I),I=1,NCLRCH)
200  FORMAT(///T60,'FIELD MEANS'///T44,4(A1,'CH(',I2,')',5X))

      WRITE(6,210)(MFANS(I),I=1,TOTMNS)
210  FORMAT(T44,4(F7.2,5X))

      RETURN
      END

```

WRT00800
 WRT00810
 WRT00820
 WRT00830
 WRT00840
 WRT00850
 WRT00860
 WRT00870
 WRT00880
 WRT00890
 WRT00900
 WRT00910
 WRT00920
 WRT00930
 WRT00940
 WRT00950
 WRT00960
 WRT00970
 WRT00980
 WRT00990
 WRT01000
 WRT01010
 WRT01020
 WRT01030
 WRT01040
 WRT01050
 WRT01060
 WRT01070
 WRT01080
 WRT01090
 WRT01100
 WRT01110
 WRT01120
 WRT01130
 WRT01140
 WRT01150
 WRT01160
 WRT01170
 WRT01180
 WRT01190
 WRT01200
 WRT01210
 WRT01220
 WRT01230
 WRT01240
 WRT01250
 WRT01260
 WRT01270
 WRT01280
 WRT01290
 WRT01300

16. SCTRPL PROCESSOR

FILE: SCTRPL

| | | |
|-------|---|----------|
| | SUBROUTINE SCTRPL (ARRAY, TOP) | SCT00010 |
| | IMPLICIT INTEGER (A-Z) | SCT00020 |
| C* | SCTRPL IS THE DRIVER FOR THE SCATTER PLOT PROCESSOR | SCT00030 |
| C* | | SCT00040 |
| C* | | SCT00050 |
| | DATA LIMIT/12000/ | SCT00060 |
| C | INCLUDE COMM1.LIST | SCT00070 |
| C | INCLUDE COMM12.LIST | SCT00080 |
| | COMMON/INFORM/NOCLS2, NOSUB2, NOFET2, VARSZ2, TOTVT2, NOFLD2, | SCT00090 |
| | * AVAR2, COVAR2, CLSID2, SUPNO2, SURDS2, FLOSV2, VERTX2, | SCT00100 |
| | * FETVC2(30), SUMVC2(75), SURPTH(75), CLSVC2(60), | SCT00110 |
| | * KEFPTS(60), NOGRP, GRPNAM(60), GRPDEX(61), | SCT00120 |
| | * GRPCHK(61), GROUPS(124) | SCT00130 |
| | COMMON/SCATTER/RSCALE, XYSCLF, CLRVEC(30), NCLPCH, CLRKEY, LOG, | SCT00140 |
| | * FREQ, XMAX, YMAX, XMIN, YMIN, HCKGND, XHI, XLO, YLO, XSIZ, | SCT00150 |
| | * YHI, YSIZ, NPINS, SYMPTX(32), RMATRX(60), RVEC(30), NRVCHN, NOFEAT | SCT00160 |
| | * SCALKY, MENADR, FLDAOR, PNTADR, IDADR, NC, RMFEAT, BMCOMB | SCT00170 |
| | * NOVEC, TOTMNS, SIZE, DRMID, DRMID1, DRMCLW, DRMCRI, DRMTNS, DRMTN1, | SCT00180 |
| | * DRMCNT, DRMCT1, DRMVEC, DRMVC1, VECTR1, DATA1, NVEC, NOREAD, LREAD | SCT00190 |
| | * DRMPTR, DRMPT1, FETVEC(16), DRMPLT, CSCALE | SCT00200 |
| | * NOSUB | SCT00210 |
| CSEND | DIMENSION ARRAY(1), BUFF(12000) | SCT00220 |
| C | CALL SFT11 (ARRAY(1), ARRAY(1), BUFF(1)) | SCT00230 |
| C | | SCT00240 |
| C | COMPUTE ADDRESSES | SCT00250 |
| C | | SCT00260 |
| C | 10 CALL SETADR (&20, &30, TOP, BUFF, LIMIT) | SCT00270 |
| C | | SCT00280 |
| C* | SCATTR IS THE MAIN DRIVER FOR CREATING THE SPECTRAL PLOTS | SCT00290 |
| C* | | SCT00300 |
| C | 20 CALL SCATTR (ARRAY (FLDAOR), ARRAY (VERTX2), ARRAY (VECTR1), ARRAY (MENADR | SCT00310 |
| | *), BUFF(1), BUFF(1), ARRAY (DATA1), TOP, LIMIT, BUFF(1)) | SCT00320 |
| C | | SCT00330 |
| C | PROCESS ANOTHER FILE | SCT00340 |
| C | | SCT00350 |
| C | GO TO 10 | SCT00360 |
| C | | SCT00370 |
| C | | SCT00380 |
| C | SEND* CARD | SCT00390 |
| C | | SCT00400 |
| C | 30 READ (21, 100) CARD | SCT00410 |
| C | 100 FORMAT (A4) | SCT00420 |
| | | SCT00430 |
| | | SCT00440 |
| | | SCT00450 |
| | RETURN | SCT00460 |
| | END | SCT00470 |

FILE: CLRCOD

```

SUBROUTINE CLRCOD(IB,MEANS,IDATA,IPOSTN,II)
C
C   IMPLICIT INTEGER (A-Z)
C   REAL MEANS(1)
C
C   INCLUDE CMRK1.LIST
C   INCLUDE CMRK12.LIST
COMMON/INFORM/NOCLS2,NOSUB2,NOFET2,VARSZ2,TOTVT2,NOFLD2,
*   AVAR2,COVAR2,CLSTD2,SUBNO2,SURDS2,FLDSV2,VERTX2,
*   FETVC2(30),SUBVC2(75),SURPTR(75),CLSV2(60),
*   KEPPTS(60),NOGRP,GRPNAM(60),GRPDEX(61),
*   GRPCHK(61),GROUPS(124)
COMMON/SCATTER/RSCALF,XYSCLF,CLRVEC(30),NCLRCH,CLRKEY,LOG,
*   FREQ,XMAX,YMAX,XMIN,YMIN,RCKGND,XHI,XLO,YLO,XSIZ,
*   YHI,YSIZ,NHINS,SYMTX(32),BMATRX(60),RVEC(30),NRVCHN,NOFEAT
*   ,SCALKY,MENADR,FLDADR,PNTADR,IDADR,NC,BMFEAT,BMCOMB
*   ,NOVEC,TOTMNS,SIZE,DRMID,DRMID1,DRMCLR,DRMCR1,DRMTNS,DRMTN1,
*   DRMCNT,DRMCT1,DRMVEC,DRMVC1,VECTR1,DATA1,NVEC,NOREAD,LREAD
*   ,DRMPTR,DRMPT1,FETVEC(16),DRMPLT,CSCALE
*   ,NOSUH
CSEND
C
C   DIMENSION IDATA(1)
C   LOGICAL*1 LDUM(4),LLDUM(4)
C   EQUIVALENCE (IDUM,LDUM(1)),(IIDUM,LLDUM(1))
C
C   IF (CLRKEY .NE. 3) GO TO 50
C   COLOR CODES (RADIANCE VALUES) ARE COMING FROM N-DIM HIST FILE
C   COLADR = DRMCLR + NVEC*(II-1) + IB - 1
C   CALL RREAD(COLADR,CODE,1,ISTAT1)
105 IF (ISTAT1 .EQ. 1) GO TO 105
C
C   IIDUM=0
C   IDUM=CODE
C   DO 10 I=1,NC
C   III=XSIZ*(I-1)+IPOSTN
C   LLDUM(4)=LDUM(I)
10 IDATA(III)=IIDUM
C
C   RETURN
C
C   COLOR CODES (STAT MEANS OR USER INPUT) ARE STORED IN CORE UNPACKED
C
50 IDADR = DRMID + NVEC*(II-1) + IB - 1
CALL RREAD(IDADR,IDNUM,1,ISTAT1)
55 IF (ISTAT1 .EQ. 1) GO TO 55
DO 80 I=1,NC
III = IPOSTN + XSIZ*(I-1)
JJ = (IDNUM-1)*NC + I
80 IDATA(III) = MEANS(JJ) + 0.5
RETURN
END

```

CLR00010
 CLR00020
 CLR00030
 CLR00040
 CLR00050
 CLR00060
 CLR00070
 CLR00080
 CLR00090
 CLR00100
 CLR00110
 CLR00120
 CLR00130
 CLR00140
 CLR00150
 CLR00160
 CLR00170
 CLR00180
 CLR00190
 CLR00200
 CLR00210
 CLR00220
 CLR00230
 CLR00240
 CLR00250
 CLR00260
 CLR00270
 CLR00280
 CLR00290
 CLR00300
 CLR00310
 CLR00320
 CLR00330
 CLR00340
 CLR00350
 CLR00360
 CLR00370
 CLR00380
 CLR00390
 CLR00400
 CLR00410
 CLR00420
 CLR00430
 CLR00440
 CLR00450
 CLR00460
 CLR00470
 CLR00480
 CLR00490
 CLR00500
 CLR00510
 CLR00520
 CLR00530
 CLR00540

SERIAL PAGE IS
 OF POOR QUALITY

FILE: CLRKYS

```

SUBROUTINE CLRKYS(XSIZ, IDATA, NOSUB2, CH, MEANS, NC)
C
C*
C*
C*
C
CLRKYS ADDS THE COLOR KEYS TO A UNIVERSAL FORMAT TAPE
THE COLORS ARE OUTPUT AS SQUARES IMAGES (10x10)
C
IMPLICIT INTEGER (A-Z)
REAL MEANS(NC, NOSUB2)
C
DIMENSION IDATA(XSIZ, CH)
C
LSTLIN = 0
LINE = 0
TOTKEY = 0
NKEYS = XSIZ/11
NOKEY = NOSUB2
C
90 DO 100 J=1, CH
C*
C*
C*
WRITE A SCAN LINE OF ZEROS - USED FOR SEPARATING THE THE COLORS
C
DO 100 I=1, XSIZ
100 IDATA(I, J) = 0
C
CALL WRTLN(IDATA, LSTLIN)
LINE = LINE + 1
C
110 IF (NKEYS .LE. NOKEY) NOKEY = NKEYS
KK = 0
C
DO 150 I=1, NOKEY
TOTKEY = TOTKEY + 1
DO 140 J=1, NC
DO 130 K=1, 10
KK = (I-1)*11 + K
130 IDATA(KK, J) = MEANS(J, TOTKEY) + 0.5
140 CONTINUE
C*
C*
C*
WRITE A SCAN LINE OF COLORS
C
150 CONTINUE
NOKEY = NOSUB2 - TOTKEY
C
DO 160 I=1, 10
IF (NOKEY .LE. 0 .AND. I .EQ. 10) LSTLIN = -1
160 CALL WRTLN(IDATA, LSTLIN)
LINE = LINE + 10
IF (NOKEY .LE. 0) GO TO 170
GO TO 90
C
170 CONTINUE
C
WRITE(6, 200) LINE
200 FORMAT(/154, 'COLOR KEYS = ', I4, ' LINES')
C
RETURN
END
```

FILE: CINTER

| | | |
|-----|---|-----------|
| C | SUBROUTINE CINTER(1B, IDATA, IPOSTN, II, COUNTR) | CNT000010 |
| C | CINTER STORE THE FREQUENCY COUNT IN IDATA | CNT000020 |
| C | THE FREQUENCY IS STORED ON DRUM | CNT000030 |
| C | IMPLICIT INTEGER (A-Z) | CNT000040 |
| C | INCLUDE CMH12.LIST | CNT000050 |
| | COMMON/SCITF/RSCALE, XYSCL, CLRVEC(30), NCLRCH, CLRKEY, LOG, | CNT000060 |
| | * FREQ, XMAX, YMAX, XMIN, YMIN, HCKGND, XHI, XLO, YLO, XSIZ, | CNT000070 |
| | * YHI, YSIZ, NRINS, SYMTX(32), RMATRX(60), BVEC(30), NBVCHN, NOFEAT | CNT000080 |
| | * , SCALY, MENADR, FLOADR, PNTADR, IDADR, NC, BMFEAT, BMCOMB | |
| | * , NOVEC, TOTINS, SIZE, DKMID, DRMID1, DRMCLR, DRMCRI, DRMTNS, DRMTN1, | |
| | * DRMCNT, DRMCT1, DRMVEC, DRMVC1, VECTR1, DATA1, NVEC, NOREAD, LREAD | |
| | * , DRMPTR, DRMPT1, FETVEC(16), DRMPLT, CSCALE | |
| | * , NOSUB | |
| C | SEND | CNT000100 |
| C | DIMENSION IDATA(1) | CNT000110 |
| C | COMPUTE DRUM ADDRESSES | CNT000120 |
| C | CTRADR = NVEC * (II-1) + IR + DRMCNT - 1 | CNT000130 |
| | CALL PREAD(CTRADR, COUNTR, 1, ISTAT1) | CNT000140 |
| 105 | IF (ISTAT1 .EQ. 1) GO TO 105 | CNT000150 |
| C | THE VECTOR COUNTER IS THE LAST CHANNEL | CNT000160 |
| C | I = XSIZ * NC + IPOSTN | CNT000170 |
| | IF (COUNTR .GT. 255) COUNTR = 255 | CNT000180 |
| | IDATA (I) = COUNTR | CNT000190 |
| C | RETURN | CNT000200 |
| | END | CNT000210 |
| | | CNT000220 |
| | | CNT000230 |
| | | CNT000240 |
| | | CNT000250 |
| | | CNT000260 |
| | | CNT000270 |

FILE: LINPLT

```

C      SUBROUTINE LINPLT
C      LINPLT CREATES THE PIXEL FREQUENCY PLOT ON DRUM AND LATER PRINTS
C      THE IMAGE ON THE LINE PRINTER
C      THERE ARE 3 ENTRIES :
C      1. SUBROUTINE LINPLT - COMPUTES THE SCALES
C      2. ENTRY STOPTS      - COMPUTES THE POSITION OF THE PIXEL ON THE
C                           PLOT (DRUM ADDRESS)
C                           *** MUST BE CALLED FOR EVERY PIXEL ***
C      3. ENTRY PRTPLOT    - PRINTS THE PLOT ON THE LINE PRINTER
C                           *** PIXEL MUST BE POSITIVE ***
C
C      IMPLICIT INTEGER (A-Z)
C      REAL XSCALE,XSHFT,YSCALE,YSHFT,SCALEY,SCALEX,SHFTY,SHFTX
C      REAL SUM,COUNT,LOG2
C
C      DIMENSION YAXIS(11),XAXIS(11)
C
C      INCLUDE COMRKA.LIST
C      INCLUDE COMRKA.LIST
C      INCLUDE COMRKA.LIST
C      DIMENSION HED1(15),HED2(15),DATE(3),COMENT(15)
C      EQUIVALENCE (HED1(1),HEAD(1)),(DATE(1),HEAD(22)),
C      (HED2(1),HEAD(30)),(COMENT(1),HEAD(48))
C      2 COMMON/GLORAL/HEAD(63),MAPTAP,DATE,SAVTAP,BMFILE,BMKEY,
C      HISFIL,HISKEY,TRFORM,ERIPTP,ERPKEY,MAPUNT,NOFILE,
C      DRUMAD,DRUMDS,PAGSIZ,DATFIL,STAFIL,ASAV,ASAVFL
C      NHSTUN,NHSTFI,SCTHUN,MAPFIL
C      DOTUNT,DOTFIL,NCHPAS,TRNSFL,BMTRFL,HISTFL,PCMUNT,
C      CRDUNT,PRUNT,RANDIO
C      COMMON/SCATTER/XSCALE,XYSCLF,CLHVEC(30),NCLRCH,CLRKEY,LOG,
C      FREQ,XMAX,YMAX,XMIN,YMIN,BCKGNU,XHI,XLO,YLO,XSIZ,
C      YHI,YSIZ,NRINS,SYMTX(32),BMATRX(60),BVEC(30),NBVCN,NOFEAT
C      SCALKY,MENADH,FLUADH,PNTADH,IDADR,NC,HMFEAT,RMCOMR
C      NOVEC,TOTINS,SIZF,DRMID,DRMID1,DRMCLR,DRMCR1,DRMTNS,DRMTN1,
C      DRMCNT,DRMCT1,DRMVEC,DRMVC1,VECTR1,DATA1,NVEC,NOHEAD,LREAD
C      DRMPTR,DRMPT1,FETVEC(16),DRMPLT,CSCALE
C      NOSUB
C
CSEND
C
C      DATA BLANK/0,0/
C      LOG2 = ALOG10(2.0)
C      MAXSUM = 1
C      IF (XYSCLF .EQ. 0) GO TO 70
C
C      DATA IS RESCALED TO 101 BINS
C
C      XSIZ = XSIZ
C      YSIZ = YSIZ
C      IF (XSIZF .GT. 101) XSIZ = 101
C      IF (YSIZF .GT. 101) YSIZ = 101
C
C      RANGES FOR THE X-AXIS
C
C      NSIZ = XSIZ / 10 + 1
C      XSCALE = FLOAT(XLO-XHI)/(XSIZ - 1)
C      XSHFT = FLOAT(XSIZ*XHI-XLO)/(XSIZ-1)
C      DO 50 I=1,NSIZ
C      XAXIS(NSIZ-I+1) = (10*I-9)*XSCALE*XSHFT + .501
C      CONTINUE
C      SCALEY = FLOAT(1-YSIZ)/YHI-YLO
C      SHFTY = -YHI*SCALEY + 1.0
C      SCALEX = FLOAT(1-XSIZ)/XHI-XLO
C      SHFTX = -XHI*SCALEX + 1.0
C
C      RANGES FOR THE Y-AXIS
C
C      ISIZ = YSIZ / 10 + 1
C      YSCALE = FLOAT(YLO-YHI)/(YSIZ-1)
C      YSHFT = FLOAT(YSIZ*YHI-YLO)/(YSIZ-1)
C      DO 60 I=1,ISIZ
C      YAXIS(ISIZ-I+1) = (10*I-9)*YSCALE + YSHFT + .501
C      CONTINUE
C      RETURN
C
C      DATA IS NOT RESCALED

```

L N00010
 L N00020
 L N00030
 L N00040
 L N00050
 L N00060
 L N00070
 L N00080
 L N00090
 L N00100
 L N00110
 L N00120
 L N00130
 L N00140
 L N00150
 L N00160
 L N00170
 L N00180
 L N00190
 L N00200
 L N00210
 L N00220
 L N00230
 L N00240
 L N00250
 L N00260
 L N00270
 L N00280
 L N00290
 L N00300
 L N00310
 L N00320
 L N00330
 L N00340
 L N00350
 L N00360
 L N00370
 L N00380
 L N00390
 L N00400
 L N00410
 L N00420
 L N00430
 L N00440
 L N00450
 L N00460
 L N00470
 L N00480
 L N00490
 L N00500
 L N00510
 L N00520
 L N00530
 L N00540
 L N00550
 L N00560
 L N00570
 L N00580
 L N00590
 L N00600
 L N00610
 L N00620
 L N00630
 L N00640
 L N00650
 L N00660
 L N00670
 L N00680
 L N00690
 L N00700
 L N00710
 L N00720
 L N00730
 L N00740
 L N00750
 L N00760
 L N00770
 L N00780
 L N00790

FILE: LINPLT

```

70  DSIZ = 11
    ISIZ = 11
    XSIZ = 101
    YSIZ = 101
    NO 75 1=1.DSIZ
    YAXIS(1) = YLO + (1-1)*10
75  XAXIS(1) = XLO + (1-1)*10
    RETURN

*****

C      ENTRY STOPST(COUNT,LINE,SAMPLE)
      REAL SAMPLE,LINE

C      COMPUTE POSITION ON GRAPH FOR (SAMPLE,LINE)

      IF (XYSCL.EQ. 1) GO TO 110
      YPOINT = YSIZE - (LINE - YLO)
      XPOINT = SAMPLE-XLO

C      IF (YPOINT .GT. 0) GO TO 80
      YPT = YPT + 1
      RETURN
80  IF (YPOINT .LE. 101) GO TO 90
      YPT = YPT + 1
      RETURN
90  IF (XPOINT .GE. 0) GO TO 100
      XPT = XPT + 1
      RETURN
100 IF (XPOINT .LE. 100) GO TO 115
      XPT = XPT + 1
      RETURN
110 CONTINUE
      YPOINT = LINE*SCALEY + SHFTY + .501
      XPOINT = XSIZE - (SAMPLE*SCALEX + SHFTX + .501) + 1
      IF (XPOINT .LT. 0) XPOINT = 0
      IF (XPOINT .GE. XSIZE) XPOINT = XSIZE - 1
      IF (YPOINT .LT. 1) YPOINT = 1
      IF (YPOINT .GT. YSIZE) YPOINT = YSIZE
115 XYADR = (YPOINT - 1)*XSIZE + XPOINT + DRMPLT

C      LOG X BASE 2 OF THE FREQUENCY

      IF (LOG .NE. 1) GO TO 117
      CALL RREAD(XYADR,SUM,1,1,ISTAT1)
122 IF (ISTAT1.EQ. 1) GO TO 122
      COUNT = COUNT
      SUM1 = SUM1 + ALOG10(COUNT) / LOG2
      CALL RWRITE(XYADR,SUM1,1,1,ISTAT2)
121 IF (ISTAT2.EQ. 1) GO TO 121
      SUM = SUM1 + 0.5
      GO TO 125
117 CONTINUE
      CALL RREAD(XYADR,SUM,1,1,ISTAT1)
120 IF (ISTAT1.EQ. 1) GO TO 120

C      TOTAL NO. OF OCCURRENCES FOR ALL THE DATA VECTORS ASSIGNED
      TO THIS (SAMPLE,LINE) POSITION

      SUM = SUM + COUNT
      CALL RWRITE(XYADR,SUM,1,1,ISTAT2)
123 IF (ISTAT2.EQ. 1) GO TO 123

C      SAVE THE LARGEST NO. OF OCCURRENCES

125 CONTINUE
      IF (SUM .GT. MAXSUM) MAXSUM = SUM
      RETURN

*****

C      ENTRY PRTPLT(PNTR,PNTRS)

      REAL PNTRS(1)
      DIMENSION HINS(16),PNTR(1),PNTR1(101)

C      IF (LOG .NE. 1) GO TO 129
      CALL RREAD(DRMPLT,PNTRS,10201,ISTAT3)
126 IF (ISTAT3.EQ. 1) GO TO 126

```

FILE: LINPLT

```

C
C      IF FREQ. LESS THAN ZERO. SET THE FREQ. TO 1
      DO 124 I=1,10201
      IF (PNTRS(I)) .EQ. 0.0) GO TO 127
      IF (PNTRS(I)) .GE. 1.0) GO TO 127
      PNTR(I) = 1
      GO TO 128
127  PNTR(I) = PNTRS(I)      * 0.5
128  CONTINUE
      GO TO 131
129  CONTINUE
      CALL READ(DMPLT,PNTR,10201,ISTAT3)
130  IF (ISTAT3 .EQ. 1) GO TO 130
131  CONTINUE

C
C      SET BIN LEVELS
      CALL SETMRG(66,0.66)
      WRITE(6,HEAD)
      WRITE(6,135)
135  FORMAT(//152,' PIXEL FREQUENCY SCATTER PLOT')
      IF (NBINS .GT. MAXSUM) NBINS = MAXSUM
      RANGE = MAXSUM / NBINS
      IF (MOD(MAXSUM,NBINS) .NE. 0) RANGE = RANGE + 1
      DO 140 I=1,NBINS
140  BINS(I) = RANGE*I
      WRITE(6,150) (BINS(I),I=1,NBINS)
150  FORMAT(//17X,1H1,2X,13,9(3X,13))
      NB = NBINS + 1
      NBS = 2*NBINS
      DO 155 JJ=1,4
      WRITE(6,160) ((SYMMTX(J),I=1,6),J=1,NBINS)
      WRITE(6,163) ((SYMMTX(J),I=1,6),J=NB,NBS)
163  FORMAT(1H+.16X,96A1)
155  CONTINUE
160  FORMAT(17X,96A1)
165  WRITE(6,165)
165  FORMAT(//)
      DO 200 J=1,YSIZE
      KK = (J-1)*XSIZE + 1
      KKK = KK + YSIZE - 1
      DO 180 I=1,XSIZE
      II = (J-1)*XSIZE + I
      IF (PNTR(II) .NE. 0) GO TO 170
      PNTR(II) = BLANK
      PNTR(II) = BLANK
      GO TO 180
170  BINLEV = PNTR(II) / RANGE
      IF (MOD(PNTR(II),RANGE) .NE. 0) BINLEV = BINLEV + 1
      PNTR(II) = SYMTX(BINLEV)
      PNTR(II) = SYMTX(BINLEV+NBINS)
180  CONTINUE

C
C      PRINT A LINE
      L = 11 - J/10
      IF (MOD(J,10) .EQ. 1) GO TO 190
      WRITE(6,143)
143  FORMAT(16X,1H-)
      WRITE(6,145) (PNTR(K),K=KK,KKK)
      WRITE(6,145) (PNTR(K),K=1,101)
145  FORMAT(1H+.T17,101A1)
      GO TO 200
190  WRITE(6,195) YAXIS(L)
195  FORMAT(10X,15,1X,1H+)
      WRITE(6,197) (PNTR(K),K=KK,KKK)
      WRITE(6,197) (PNTR(K),K=1,101)
197  FORMAT(1H+.T17,101A1)
200  CONTINUE

C
C      PRINT X-AXIS SCALES
      WRITE(6,220) (XAXIS(I),I=1,NSIZ)
220  FORMAT(1H+.14X, 10(1H+.9(1H-),1H+/14X,11(13,7X))

C
      TOTPTS = XPT + YPT
      IF (TOTPTS .EQ. 0) RETURN
      WRITE(6,225) TOTPTS

```

```

L NO 590
L NO 600
L NO 610
L NO 620
L NO 630
L NO 640
L NO 650
L NO 660
L NO 670
L NO 680
L NO 690
L NO 700
L NO 710
L NO 720
L NO 730
L NO 740
L NO 750
L NO 760
L NO 770
L NO 780
L NO 790
L NO 800
L NO 810
L NO 820
L NO 830
L NO 840
L NO 850
L NO 860
L NO 870
L NO 880
L NO 890
L NO 900
L NO 910
L NO 920
L NO 930
L NO 940
L NO 950
L NO 960
L NO 970
L NO 980
L NO 990
L NO2010
L NO2010
L NO2020
L NO2020
L NO2030
L NO2040
L NO2050
L NO2060
L NO2070
L NO2080
L NO2090
L NO2100
L NO2110
L NO2120
L NO2130
L NO2140
L NO2150
L NO2160
L NO2170
L NO2180
L NO2190
L NO2200
L NO2210
L NO2220
L NO2230
L NO2240
L NO2250
L NO2260
L NO2270
L NO2280
L NO2290
L NO2300
L NO2310
L NO2320
L NO2330
L NO2340
L NO2350
L NO2360
L NO2370

```

FILE: LINPLT

```
225 FORMAT(/// 'A TOTAL OF 140 POINTS WERE NOT DISPLAYED ON THE LINFLIN02380  
* PRINTER GRAPH. THE POINTS WERE OUT OF RANGE IN EITHER THE X DIRECTION OR Y DIRECTION')  
CALL SETMRG(66.4.62) LIN02390  
RETURN LIN02400  
END LIN02410  
LIN02420  
LIN02430
```

FILE: MATINS

```
C      SUBROUTINE MATINS(A,B,C,D,L,M)
C      MULTIPLY A BY B AND ADD D, STORE IN C
C      INTEGER B
C      DIMENSION A(L,M),B(M),C(L),D(L)
C      DO 20 I=1,L
C      SUM = 0.0
C      DO 10 K=1,M
C      10 SUM = SUM + A(I,K) * B(K)
C      20 C(I) = SUM + D(I)
C      RETURN
C      END
```

ORIGINAL PAGE IS
OF POOR QUALITY

FILE: OFFSET

| | | |
|-------|---|----------|
| | SUBROUTINE OFFSET(YSCALE,XSCALE) | OFF00010 |
| C | INTEGER BMKEY,RSCALE | OFF00020 |
| | INTEGER YSIZ,XSIZ,XHI,YHI,XLO | OFF00030 |
| | INTEGER CSCALE | OFF00040 |
| C | INCLUDE COMRK6.LIST | OFF00050 |
| C | INCLUDE CMK12.LIST | OFF00060 |
| | COMMON/GLOBAL/HEAD(63),MAPTAP,DATAP,SAVTAP,BMFILE,BMKEY, | OFF00070 |
| | HISFIL,HISKEY,ISFORM,ER1PTP,ERPKEY,MAPUNT,NUFILE, | COM00010 |
| | DRIMAD,DRMVNS,PAGSIZ,DATFIL,STAFIL,ASAV,ASAVFL | COM00020 |
| | NMSTUN,NMSTFI,SCTRUN,MAPFIL | COM00030 |
| | DOTUNT,DOTFIL,NCHPAS,TRANSF,RMTRFL,HISTFL,PCMUNT, | COM00040 |
| | CRDUNT,PHTUNT,HANDIO | COM00050 |
| | COMMON/SCATTER/SCALE,XSCALE,CLRVEC(30),NCLPCH,CLRKEY,LOG, | COM00060 |
| | FRFC,XMAX,YMAX,XMIN,YMIN,HCKGND,XHI,XLO,YLO,XSIZ, | |
| | YHI,YSIZ,NRINS,SYMMTX(32),RMATRX(60),HVEC(30),NMVCHN,NOFEAT | |
| | SCALXY,MFNADH,FLNADH,PNTADR,IDAOR,NC,RMFEAT,RMCOMB | |
| | NVVEC,TOTMNS,SIZE,DRMID,DRMID1,DRMCLN,DRMCR1,DRMTNS,DRMTN1, | |
| | DRMCT1,DRMCT1,DRMVEC,DRMVC1,VECTR1,DAT1,NVEC,NOREAD,LREAD | |
| | DRMPTR,DRMPT1,FETVEC(16),DRMPLT,CSCALE | |
| | NOSUM | |
| CSEND | | OFF00100 |
| C | DIMENSION YSCALE(YSIZ),XSCALE(XSIZ) | OFF00110 |
| C | SCALES ARE COMPUTED IN ONE OF 3 WAYS : | OFF00120 |
| C | 1) DATA HAS NOT BEEN TRANSFORMED | OFF00130 |
| C | BMKEY = 0 | OFF00140 |
| C | 2) DATA HAS BEEN TRANSFORMED AND RESCALED | OFF00150 |
| C | HFSCALE = 1 | OFF00160 |
| C | 3) DATA HAS BEEN TRANSFORMED, BUT NOT RESCALED | OFF00170 |
| C | HFSCALE = 0 | OFF00180 |
| C | CSCALE = 1 -- YSIZ AND/OR XSIZ HAS BEEN INPUT - MIN AND MAX | OFF00190 |
| C | WILL BE USED FOR HI AND LO PARAMETERS | OFF00200 |
| | | OFF00210 |
| | IF (BMKEY.EQ. 0) GO TO A0 | OFF00220 |
| | IF (HFSCALE.EQ. 1) GO TO A0 | OFF00230 |
| | IF (CSCALE.EQ. 1) GO TO A0 | OFF00240 |
| C | | OFF00250 |
| | XHIGH = XMAX | OFF00260 |
| | XLOW = XMIN | OFF00270 |
| | YHIGH = YMAX | OFF00280 |
| | YLOW = YMIN | OFF00290 |
| | GO TO 90 | OFF00300 |
| C | | OFF00310 |
| C | 80 XHIGH = XHI | OFF00320 |
| | XLOW = XLO | OFF00330 |
| | YHIGH = YHI | OFF00340 |
| | YLOW = YLO | OFF00350 |
| C | | OFF00360 |
| C | 90 CONTINUE | OFF00370 |
| | XINC = (XHIGH - XLOW) / FLOAT(XSIZ-1) | OFF00380 |
| | YINC = (YHIGH - YLOW) / FLOAT(YSIZ-1) | OFF00390 |
| | XSCALE(1) = XLOW | OFF00400 |
| | YSCALE(1) = YLOW | OFF00410 |
| | DO 100 I=2,XSIZ | OFF00420 |
| 100 | XSCALE(I) = XSCALE(I-1) + XINC | OFF00430 |
| | DO 110 I=2,YSIZ | OFF00440 |
| 110 | YSCALE(I) = YSCALE(I-1) + YINC | OFF00450 |
| | RETURN | OFF00460 |
| | END | OFF00470 |
| | | OFF00480 |
| | | OFF00490 |
| | | OFF00500 |
| | | OFF00510 |

FILE: RESCLE

```

C      SURROUTINE RESCLE(DATA,SWTCH,NVECT)
C
C      RESCLE RESCALES THE TRANSFORMED DATA
C      SWITCH = 0 RESCALE THE ENTIRE ARRAY
C      SWITCH = 1 RESCALE ONE PIXEL AND RETURN
C
C      INTEGER BMCOMB,SWTCH
C      INTEGER XHI,XLO,YHI,YLO
C      REAL MAX(2),MIN(2),REAL(2),RANGE(2),DATA(2,NVECT)
C
C      INCLUDE CMRK12.LIST
C      COMMON/SCATTER/RSCL,XYSCLE,CLPVEC(30),NCLRCH,CLRKEY,LOG,
C      * FREQ,XMAX,YMAX,XMIN,YMIN,BCKGND,XHI,XLO,YLO,XSIZ,
C      * YHI,YSIZ,NRINS,SYMTX(32),RMATX(60),BVEC(30),NBVCHN,NOFEAT
C      * SCALKY,MENADW,FLDADW,PNTADR,IDADR,NC,RMFEAT,BMCOMB
C      * NOVEC,TOTMNS,SIZF,DRMID,DRMID1,DRMCLK,DRMCRI,DRMTNS,DRMTN1,
C      * DRMCNT,DRMCT1,DRMVEC,DRMVC1,VECTR1,DATA1,NVEC,NOREAD,LREAD
C      * DRMPTR,DRMPT1,FETVEC(16),DRMPLT,CSCALE
C      * NOSUB
C
C      MAX(1) = XMAX
C      MAX(2) = YMAX
C
C      MIN(1) = XMIN
C      MIN(2) = YMIN
C      RANGE(1) = XHI - XLO
C      RANGE(2) = YHI - YLO
C      DO 200 J=1,NVECT
C      DO 100 I=1,BMCOMB
C
C      REAL(I) = RANGE(I) / (MAX(I)-MIN(I))
C      100 DATA(I,J) = REAL(I)*ABS(MIN(I) - DATA(I,J))
C      IF (SWTCH .EQ. 1) RETURN
C
C      200 CONTINUE
C      RETURN
C      END

```

RES00010
 RES00020
 RES00030
 RES00040
 RES00050
 RES00060
 RES00070
 RES00080
 RES00090
 RES00100
 RES00110
 RES00120

RES00140
 RES00150
 RES00160
 RES00170
 RES00180
 RES00190
 RES00200
 RES00210
 RES00220
 RES00230
 RES00240
 RES00250
 RES00260
 RES00270
 RES00280
 RES00290
 RES00300
 RES00310
 RES00320

ORIGINAL PAGE IS
 OF POOR QUALITY

FILE: SCATTR

```

SUBROUTINE SCATTR(FIELDS,VERTEX,TNSDAT,MEANS,PLOT,PNTR,IDATA,TOP, SCA00010
  LIMIT,BUFF) SCA00020
C SCA00030
C* SCATTR SETS UP THE LOGIC FOR CREATING THE SPECTRAL PLOTS SCA00040
C SCA00050
  IMPLICIT INTEGER (A-Z) SCA00060
  REAL XSCALE,YSCALE,XLOWER,XUPPER,YUPPER,TNSDAT(2,1) SCA00070
  REAL YLOWER,LINE,SAMPLE SCA00080
  LOGICAL SWITCH(4) SCA00090
C SCA00100
  DIMENSION RIRF(1) SCA00110
  DIMENSION XSCALE(200),YSCALE(200) SCA00120
  DIMENSION IRG(4),TEN(4) SCA00130
  DIMENSION FIELDS(4,1),VERTEX(2,1),PLOT(1),MEANS(1) SCA00140
  *PNTR(1),IDATA(XSIZE,NC) SCA00150
  DIMENSION LINADR(4),LINE(4) SCA00160
  DIMENSION COVAR(465) SCA00170
C SCA00180
C SCA00190
C SCA00200
  INCLUDE COMRK1.LIST SCA00210
  INCLUDE COMRK6.LIST SCA00220
  INCLUDE COMRK12.LIST SCA00230
  COMMON/INFORM/NOCLS2,NOSUB2,NOFET2,VARSZ2,TOTVT2,NOFLD2, SCA00240
  *AVAR2,COVAR2,CLS1D2,SURNO2,SUBDS2,FLDSV2,VERTX2, SCA00250
  *FETVC2(30),SUBVC2(75),SUBPTR(75),CLSV2(60), SCA00260
  *KEPPTS(60),NOGRP,GHPNAM(60),GRPDEX(61), SCA00270
  *GRPCHK(61),GROUPS(124) SCA00280
  COMMON/GLOBAL/HEAD(63),MAPTAP,DATEP,SAVTAP,BMFILE,BMKEY, SCA00290
  *HISFIL,HISKEY,TRFORM,ERIPTR,ERPKEY,MAPUNT,NOFILE, SCA00300
  *DRUMAD,DRMADS,PAGSIZ,DATEFIL,STAFIL,ASAV,ASAVFL SCA00310
  *NHSTUN,NHSTFI,STRUN,MAPEIL SCA00320
  *DOTUNT,DOTFIL,NCHPAS,TRNSFL,BMTRFL,HISTFL,PCHUNT, SCA00330
  *CRDUNT,PRTUNT,PANDIO SCA00340
  COMMON/SCATTER/XSCALE,XYSIZE,CLPVEC(30),NCLRCH,CLPKEY,LOG, SCA00350
  *FREQ,XMAX,YMAX,XMIN,YMIN,BCKGND,XHI,XLO,YLO,XSIZ, SCA00360
  *YHI,YSIZ,NRINS,SYMMTX(32),RMATRX(60),HVEC(30),NBVCHN,NOFEAT SCA00370
  *SCALKY,MENADR,FLDADR,PNTADR,IDADR,NC,BMFEAT,BMCOMB SCA00380
  *NOVEC,TOTMNS,SIZE,DRMID,DRMID1,DRMCLR,DRMCR1,DRMTNS,DRMTN1, SCA00390
  *DRMCNT,DRMCT1,DRMVEC,DRMVC1,VECTR1,DATA1,NVEC,NOREAD,LREAD SCA00400
  *DRMPTR,DRMPT1,FETVEC(16),DRMPLT,CSCALE SCA00410
  *NOSUB SCA00420
CSEND SCA00430
C SCA00440
C SCA00450
  READ REC 2 FORM N-DIM HIST FILE SCA00460
  READ(NHSTUN)CLSV2(1),(SURVC2(1),I=1,NOSUB),((FIELDS(I,J), SCA00470
  * I=1,4),J=1,NOFLD2),((VERTEX(I,J),I=1,2),J=1,TOTVT2) SCA00480
C SCA00490
C SCA00500
  ARE COLOR CODES COMING FROM STAT FILE SCA00510
  IF (CLPKEY.NE. 1) GO TO 90 SCA00520
  IF (NOFEAT.NE. 0) GO TO 80 SCA00530
C SCA00540
C SCA00550
  DEFAULT CHANNELS ARE CHANNELS FROM N-DIM HIST FILE SCA00560
  DO 60 I=1,NOFET2 SCA00570
60 FETVEC(I) = FETVC2(I) SCA00580
C SCA00590
C SCA00600
  EXTRACT JUST MEANS FROM STAT FILE SCA00610
  * MEANS,COVAR,0) SCA00620
C SCA00630
C SCA00640
  WRITE OUT SAVED TRAINING/TEST FIELDS SCA00650
90 CALL WRTFLD(FIELDS,VERTEX,NOFLD2,2,CLSV2,SUBVC2) SCA00660
C SCA00670
C SCA00680
  READ N-DIM HIST FILE AND STORE INFO ON DRUM SCA00690
  CALL STOFIL(LIMIT,MEANS,BUFF) SCA00700
  DRMVC1 = DRMVEC SCA00710
  NVEC = NVEC SCA00720
  DO 93 II=1,NOREAD SCA00730
  IF (II.EQ. NOREAD) NVEC = LREAD SCA00740
  NWORDS = NVEC*SIZE SCA00750
  CALL RRFAD(DRMVC1,PLOT,NWORDS,STAT) SCA00760
  DRMVC1 = DRMVC1 + NWORDS SCA00770
91 IF (STAT.EQ. 1) GO TO 91 SCA00780
C SCA00790
  APPLY TRANSFORMATION SCA00790

```

FILE: SCATTR

```

C      IF (BMKEY .NE. 0) CALL TNSFER(PLOT,TNSDAT,NVECT,II)
C      NO TRANSFORMATION APPLIED
C      IF (BMKEY .EQ. 0) CALL UNPCKV(PLOT,TNSDAT,NVECT)
C      SORT VECTORS IN DECENDING ORDER
C      ICOL = 2
C      CALL SORTVC(TNSDAT,PNTR,ICOL,NVECT,IBG,IEN,II)
C      DRMWD = 2*NVECT
C      CALL RWRITE(DRMVN,TNSDAT,DRMWD,ISTAT1)
92  IF (ISTAT1 .EQ. 1) GO TO 92
C      DRMTN1 = DRMTN1 + DRMWD
C      CALL RWRITE(DRMPT1,PNTR,NVECT,ISTAT2)
94  IF (ISTAT2 .EQ. 1) GO TO 94
C      DRMPT1 = DRMPT1 + NVECT
93  CONTINUE
C      IF (RSCALE .EQ. 0) GO TO 97
C      IF (SCALKY .NE. 2) GO TO 97
C      DRMTN1 = DRMTN1
C      NTNSVC = (TOP - VECTRI) / 2
C      IF (NOVEC .LT. NTNSVC) GO TO 95
C      NREAD = NOVEC/NTNSVC
C      IF (MOD(NOVEC,NTNSVC) .NE. 0) NREAD = NREAD
C      LSREAD = (MOD(NOVEC,NTNSVC))
C      IF (LSREAD .EQ. 0) LSREAD = NTNSVC
C      GO TO 99
95  NREAD = 1
C      NTNSVC = NOVEC
C      LSREAD = NTNSVC
98  DO 99 I=1,NREAD
C      IF (II .EQ. NREAD) NTNSVC = LSREAD
C      DRMWD = 2 * NTNSVC
C      CALL RREAD (DRMTN1,TNSDAT,DRMWD ,ISTAT2)
101 IF (ISTAT2 .EQ. 1) GO TO 101
C      CALL RESCLE(TNSDAT,0,NTNSVC)
C      CALL RWRITE(DRMVN,TNSDAT,DRMWD ,ISTAT3)
100 IF (ISTAT3 .EQ. 1) GO TO 100
C      DRMTN1 = DRMTN1 + DRMWD
99  CONTINUE
C      COMPUTE TAPE CO/ORDINATES
C      97 CONTINUE
C      CALL OFFSET(YSCALE,XSCALE)
C      WRITE TAPE PARAMETERS
C      WRITE(6,300)YSIZ,XSIZ
300 FORMAT(/T51,'SCATTER PLOT TAPE PARAMETERS',//T51,'NO. OF LINES PER
C      * FILE =',I4/T51,'NO. OF SAMPLES PER LINE =',I4)
C      IF (BMKEY .EQ. 0) WRITE(6,310)XLO,YLO,XHI,YHI
310 FORMAT(/T51,'XLO =',I4,T71,'YLO =',I4/T51,'XHI =',I4,T71,'YHI =',
C      * I4)
C      IF (RMKEY .NE. 0 .AND. RSCALE .EQ. 1) WRITE(6,310)XLO,YLO,XHI,YHI
C      IF (CSCALE .EQ. 1 .AND. BMKEY .GT. 0) WRITE(6,310)XLO,YLO,XHI,YHI
C      IF (RMKEY .NE. 0) WRITE(6,320)XMIN,YMIN,XMAX,YMAX
320 FORMAT(/T50,'XMIN =',F10.5,T70,'YMIN =',F10.5/T50,'XMAX =',F10.5,
C      * T70,'YMAX =',F10.5)
C      FORMAT = 1
C      CH = NC + 1
C      DO 110 I=1,CH
110 CLRVEC(I) = 1
C      120 CALL WRTHED(CH,CLRVEC,XSIZ,FORMAT,SCTRUN)
C      LSTLIN = 0
C      IF (LOG .EQ. 1 .OR. FREQ .EQ. 1) CALL LINPLT
C      DO 125 I=1,NOREAD
125 SWITCH(I) = .TRUE.
C      CALL RREAD(DRMPTN,PNTR,NOVEC,ISTAT3)
C      I=YSIZ+1

```

ORIGINAL PAGE
OF POOR QUALITY

SCA00800
SCA00810
SCA00820
SCA00830
SCA00840
SCA00850
SCA00860
SCA00870
SCA00880
SCA00890
SCA00900
SCA00910
SCA00920
SCA00930
SCA00940
SCA00950
SCA00960
SCA00970
SCA00980
SCA00990
SCA01000
SCA01010
SCA01020
SCA01030
SCA01040
SCA01050
SCA01060
SCA01070
SCA01080
SCA01090
SCA01100
SCA01110
SCA01120
SCA01130
SCA01140
SCA01150
SCA01160
SCA01170
SCA01180
SCA01190
SCA01200
SCA01210
SCA01220
SCA01230
SCA01240
SCA01250
SCA01260
SCA01270
SCA01280
SCA01290
SCA01300
SCA01310
SCA01320
SCA01330
SCA01340
SCA01350
SCA01360
SCA01370
SCA01380
SCA01390
SCA01400
SCA01410
SCA01420
SCA01430
SCA01440
SCA01450
SCA01460
SCA01470
SCA01480
SCA01490
SCA01500
SCA01510
SCA01520
SCA01530
SCA01540
SCA01550
SCA01560
SCA01570
SCA01580

FILE: SCATTR

```

130 I=I-1
    IF (I.EQ. 1 .AND. CLRKEY.EQ. 3) LSTLIN = -1
    ILINE = I
    YUPPER = YSCALE(I)
    YLOWER = YSCALE(I-1)
C
    DO 140 K=1,CH
    DO 140 J=1,XSIZ
140 IDATA(J,K) = BCKGND
C
143 IF (ISTAT3.EQ. 1) GO TO 143
    COLLECT ALL POINTS THAT BELONG TO THIS LINE(I)
C
    THE DATA VECTORS WERE READ IN NVEC AT A TIME. EACH BLOCK OF DATA
    VECTORS HAS ITS OWN POINTER ARRAY FOR SORTING THE DATA VECTORS IN
    DESCENDING ORDER. EACH POINTER ARRAY PNTR(1...NOREAD) MUST BE
    SEARCHED FOR POINTS BELONGING TO LINE(I)
C
    DO 180 II=1,NOREAD
    K = (II-1)*NVEC
    IR = IRG(II)
145 IF (SWITCH(II)) LINADR(II) = (2*NVEC)*(II-1) + 2*IR + DRMTNS - 1
    IF (SWITCH(II)) CALL RREAD(LINADR(II),LINE(II),1,ISTAT4)
146 IF (ISTAT4.EQ. 1) GO TO 146
    IF (ILINE.NE. 1) GO TO 1465
    IF (LINE(II).LE. YUPPER) GO TO 147
1465 CONTINUE
    IF (LINE(II).LE. YUPPER .AND. LINE(II).GT. YLOWER) GO TO 147
    SWITCH(II) = .FALSE.
    IF (ILINE.NE. YSIZ) GO TO 180
    IF (LINE(II).GT. YUPPER) GO TO 147
    GO TO 180
C
    POSITION POINT IN X CO-ORDINATES
C
147 SAMADR = (2*NVEC)*(II-1) + 2*IR + DRMTNS - 2
    CALL RREAD(SAMADR,SAMPLE,1,ISTAT5)
    SWITCH(II) = .TRUE.
148 IF (ISTAT5.EQ. 1) GO TO 148
    DO 150 J=1,XSIZ
    IPOSTN = J
    XLOWER = XSCALE(J)
    XUPPER = XSCALE(J+1)
C
    IF (J.NE. 1) GO TO 149
    IF (SAMPLE.LE. XLOWER) GO TO 160
149 CONTINUE
    IF (SAMPLE.GE. XLOWER .AND. SAMPLE.LT. XUPPER) GO TO 160
150 CONTINUE
C
    GET COLOR CODES
C
160 CALL CLPCOD(IR,MEANS,IDATA,IPOSTN,II)
    CALL CTRER(IR,IDATA,IPOSTN,II,COUNTN)
    IF (LOG.EQ. 1 .OR. FREQ.EQ. 1) CALL STOPTS(COUNTN,LINE(II),
        SAMPLE)
C
    CHECK NEXT VECTOR
C
    IF (IR.EQ. IFN(II)) GO TO 180
    IF (SWITCH(II)) IR = PNTR(IR*K)
    IF (SWITCH(II)) IRG(II) = IR
    IF (SWITCH(II)) GO TO 145
C
180 CONTINUE
C
    WRITE A LINE
C
    CALL WRTLN(IDATA,LSTLIN)
200 IF (I.GT.1) GO TO 130
C
    IF (CLRKEY.EQ. 1) CALL CLRKYS(XSIZ,IDATA,NOSUB2,CH,MEANS,NC)
    IF (CLRKEY.EQ. 2) CALL CLRKYS(XSIZ,IDATA,NOSUB2,CH,MEANS,NC)
    IF (CLRKEY.EQ. 4) CALL CLRKYS(XSIZ,IDATA,NOFD2,CH,MEANS,NC)
C
    IF (LOG.EQ. 1 .OR. FREQ.EQ. 1) CALL PRTPLT(RUFF,BUFF)
C

```

SCA01590
 SCA01600
 SCA01610
 SCA01620
 SCA01630
 SCA01640
 SCA01650
 SCA01660
 SCA01670
 SCA01680
 SCA01690
 SCA01700
 SCA01710
 SCA01720
 SCA01730
 SCA01740
 SCA01750
 SCA01760
 SCA01770
 SCA01780
 SCA01790
 SCA01800
 SCA01810
 SCA01820
 SCA01830
 SCA01840
 SCA01850
 SCA01860
 SCA01870
 SCA01880
 SCA01890
 SCA01900
 SCA01910
 SCA01920
 SCA01930
 SCA01940
 SCA01950
 SCA01960
 SCA01970
 SCA01980
 SCA01990
 SCA02000
 SCA02010
 SCA02020
 SCA02030
 SCA02040
 SCA02050
 SCA02060
 SCA02070
 SCA02080
 SCA02090
 SCA02100
 SCA02110
 SCA02120
 SCA02130
 SCA02140
 SCA02150
 SCA02160
 SCA02170
 SCA02180
 SCA02190
 SCA02200
 SCA02210
 SCA02220
 SCA02230
 SCA02240
 SCA02250
 SCA02260
 SCA02270
 SCA02280
 SCA02290
 SCA02300
 SCA02310
 SCA02320
 SCA02330
 SCA02340
 SCA02350
 SCA02360
 SCA02370

FILE: SCATTR

ORIGINAL PAGE IS
OF POOR QUALITY

400 RETURN
END

SCA02380
SCA02390

~~16-15~~
306

FILE: SETADR

SUBROUTINE SETADR(*,*,TOP,BUFF,LIMIT)

SETADR COMPUTES THE ADDRESS FOR STORING THE NDIM FILE ON DRUM
AND ADDRESS FOR THE TWO ARRAYS - BUFF(LIMIT) AND ARRAY(TOP)

IMPLICIT INTEGER (A-Z)

INCLUDE COMK1.LIST
INCLUDE COMK6.LIST
INCLUDE COMK12.LIST
INCLUDE COMT12.LIST

COMMON/INFORM/NOCLS2,NOSUR2,NOFET2,VARSZ2,TOTVT2,NOFLD2,
AVAR2,COVAR2,CLS1D2,SURN02,SURDS2,FLDSV2,VERTX2,
FETVC2(30),SUBVC2(75),SUBPTR(75),CLSV2(60),
KEPPTS(60),NOGRP,GRPNAM(60),GRPDEX(61),
GRPCHK(61),GROUPS(124)
COMMON/GLOBAL/HEAD(63),MAPTAP,DATAP,SAVTAP,BMFILE,BMKEY,
HISFIL,HISKEY,TRFORM,ER1PT,ERPKEY,MAPUNT,NOFILE,
DRUMAD,DRUMDS,PAGSIZ,DATFIL,STAFIL,ASAV,ASAVFL
NHSTUN,NHSTFI,SCTRUN,MAPFIL
DOTUNT,DOTFIL,NCHPAS,TRNSFL,BMTRFL,HISTFL,PCHUNT,
CRDUNT,PRTUNT,RANDIO
COMMON/SCITFR/PSCALE,XYSCL,CLRVEC(30),NCLPCH,CLRKEY,LOG,
FREQ,XMAX,YMAX,XMIN,YMIN,BCKGND,XHI,XLO,YLO,XSIZ,
YHI,YSIZ,NRINS,SYMTX(32),RMATX(60),RVFC(30),NRVCHN,NOFEAT
SCALKY,MENADR,FLDADR,PNTADR,1DADR,NC,BMFEAT,BMCOMB
NOVEC,TOTMNS,SIZE,DRMID,DRMID1,DRMCLR,DRMCR1,DRMTNS,DRMTN1,
DRMCNT,DRMCT1,DRMVEC,DRMVC1,VECTR1,DAT1,NVEC,NOREAD,LREAD
DRMPTR,DRMPT1,FETVEC(16),DRMPLT,CSCALE
NOSTH

SCITFR IS A COMMON BLOCK LOADED ONLY WITH THE SCATTER PLOT
PROCESSOR

PSCALE - KEY INDICATING THAT THE TRANSFORMED DATA IS TO BE RESCALED
XYSCL - KEY INDICATING THAT THE PIXEL FREQUENCY PLOT DATA IS
TO BE RESCALED TO A RANGE OF 100
CLRVEC - ARRAY CONTAINING THE COLOR CHANNELS
NCLPCH - NO. OF COLOR CHANNELS
CLRKEY - KEY INDICATING THE MANNER THAT THE COLORS ARE BEING
DEFINED :

= 1 - INPUT BY STAT FILE
= 2 - USER INPUT
= 3 - RADIANCE VALUES FROM IMAGE TAPE- ON NDIM FILE
= 4 - FIELD MEANS - ON NDIM FILE

LOG - KEY INDICATING LOG(2) OF FREQUENCY IS TO BE OUTPUT ON
LINE PRINTER PLOT
FREQ - KEY INDICATING FREQUENCY IS TO BE OUTPUT ON LINE PRINT-
ER PLOT
XMAX - MAXIMUM VALUE OF 1ST COMPONENT OF TRANS. DATA
YMAX - MAXIMUM VALUE OF 2ND COMPONENT OF TRANS. DATA
XMIN - MINIMUM VALUE OF 1ST COMPONENT OF TRANS. DATA
YMIN - MINIMUM VALUE OF 2ND COMPONENT OF TRANS. DATA
BCKGND - COLOR FOR TAPE BACKGROUND
XHI - UPPER LIMIT OF SCAN LINE PARAMETER
XLO - LOWER LIMIT OF SCAN LINE PARAMETER
YHI - UPPER LIMIT OF LINE NO. PARAMETER
YLO - LOWER LIMIT OF LINE NO. PARAMETER
XSIZ - NO. OF SAMPLES PER SCAN LINE

CONTINUE
YSIZ - NO. OF LINES TO OUTPUT ON TAPE
NRINS - NO. OF MIN LEVELS OR SYMBOLS FOR PIXEL FREQ. PLOT
SYMTX - ARRAY CONTAINING SYMBOLS FOR PIXEL FREQ. PLOT
RMATX - ARRAY CONTAINING R-MATRIX
RVFC - ARRAY CONTAINING ADDITIVE VECTOR
NRVCHN - NO. OF ADDITIVE VECTOR ELEMENTS
NOFEAT - NOFET2 + NCLPCH
SCALKY - KEY INDICATING THE MANNER OF COLLECTING THE MIN AND MAX
VALUES :

= 1 - USER INPUT
= 2 - COMPUTE FROM NDIM FILE

MENADR - ADDRESS FOR STORING MEANS
FLDADR - ADDRESS FOR STORING FIELD INFO
1DADR - ADDRESS FOR STORING IDS
NC - NO. OF CHANNELS FOR COLORS TO BE OUTPUT ON TAPE
BMFEAT - NO. OF CHANNELS IN B-MATRIX
BMCOMB - NO. OF LINEAR COMB.
NOVEC - NO. OF VECTORS ON NDIM FILE
TOTMNS - NO. OF MEAN ELEMENTS

SET00010
SET00020
SET00030
SET00040
SET00050
SET00060
SET00070
SET00080
SET00090
SET00100
SET00110
SET00120
SET00130
SET00140
SET00150
SET00160
SET00170
SET00180
SET00190
SET00200
SET00210
SET00220
SET00230
SET00240
SET00250
SET00260
SET00270
SET00280
SET00290
SET00300
SET00310
SET00320
SET00330
SET00340
SET00350
SET00360
SET00370
SET00380
SET00390
SET00400
SET00410
SET00420
SET00430
SET00440
SET00450
SET00460
SET00470
SET00480
SET00490
SET00500
SET00510
SET00520
SET00530
SET00540
SET00550
SET00560
SET00570
SET00580
SET00590
SET00600
SET00610
SET00620
SET00630
SET00640
SET00650
SET00660
SET00670
SET00680
SET00690
SET00700
SET00710
SET00720
SET00730
SET00740
SET00750
SET00760
SET00770
SET00780
SET00790

FILE: SETADR

```

C*  SIZE - NO. OF WORDS FOR PACKED HISTOGRAMMED VECTOR
C*  DRMTD - BEGINNING DRUM ADDRESS FOR STORING IDS
C*  DRMTD1 - SUMMING DRUM ADDRESS FOR RETRIV. IDS
C*  DRMCLR - BEGINNING DRUM ADDRESS FOR STORING COLORS
C*  DRMCR1 - SUMMING DRUM ADDRESS FOR RETRIV. COLORS
C*  DRMTNS - BEGINNING DRUM ADDRESS FOR STORING TRANS. DATA
CONTINUE
C*  DRMTN1 - SUMMING DRUM ADDRESS FOR RETRIV. TRANS. DATA
C*  DRMCNT - BEGINNING DRUM ADDRESS FOR STORING FREQ.
C*  DRMCT1 - SUMMING DRUM ADDRESS FOR RETRIV. FREQ.
C*  DRMVFC - BEGINNING DRUM ADDRESS FOR STORING VECTORS
C*  DRMVC1 - SUMMING DRUM ADDRESS FOR RETRIV. VECTORS
C*  VECTR1 - ADDRESS IN 'ARRAY' FOR STORING TRANS. VECTOR
C*  DATA1 - SAME ADDRESS AS VECTR1 - USED FOR CREATING A SCAN LINE
C*  NVFC - NO. OF VECTORS TO READ FROM DRUM AT ONE TIME
C*  NOREAD - NO. OF READS TO DRUM
C*  LREAD - NO. OF VECTOR TO READ ON LAST DRUM READ
C*  DRMPTR - BEGINNING DRUM ADDRESS FOR STORING POINTERS
C*  FETVEC - ARRAY FOR CONTAINING FETVC2 AND CLRVEC CHANNELS
C*  DRMPLT - BEGINNING ADDRESS FOR STORING FREQ. PLOT IMAGE
C*  CSCALE - KEY INDICATING THAT XSIZ OR YSIZ PARAMETERS HAVE BEEN
          INPUT

```

CSEND

DIMENSION BUFF(1)

READ(NHSTUN,END=150) NOFLD2,NOSUR,TOTVT2,NOVEC

```

IF (CLRKEY .EQ. 1) GO TO 100
IF (CLRKEY .EQ. 2) GO TO 100
IF (CLRKEY .EQ. 3) GO TO 120
IF (CLRKEY .EQ. 4) GO TO 130

```

```

C 100 MENADR = 1
    FLDADR = MENADR + 60*NC
    VERTX2 = FLDADR + 4*NOFLD2
    DATA1 = FLDADR
    GO TO 133

```

```

C 120 FLDADR = 1
    VERTX2 = FLDADR + NOFLD2*4
    DATA1 = FLDADR
    GO TO 133

```

```

C 130 MENADR = 1
    FLDADR = MENADR + TOTMNS
    VERTX2 = FLDADR + 4*NOFLD2
    DATA1 = FLDADR

```

COMPUTE MAXIMUM NO. OF VECTORS ARRAY MAY HOLD AT ONE TIME

```

C 133 VECTR1 = DATA1
    NVFC = (TOP - DATA1) / 2
    VECTR2 = LIMIT / SIZE
    IF (VECTR2 .LT. NVFC) NVFC = VECTR2
    IF (NOVEC .LT. NVFC) GO TO 135
    NOREAD = NOVEC / NVFC
    IF (MOD(NOVEC,NVEC) .NE. 0) NOREAD = NOREAD + 1
    LREAD = MOD(NOVEC,NVEC)
    IF (LREAD .EQ. 0) LREAD = NVFC
    GO TO 140
C 135 NOREAD = 1
    NVFC = NOVEC
    LREAD = NVFC

```

ADDRESSES FOR HIGH SPEED DRUM

```

C 140 DRMVFC = DRMTAD
    DRMTD = DRMVFC + NOVEC*SIZE
    DRMCNT = DRMTD + NOVEC
    IF (CLRKEY .EQ. 3) DRMCLR = DRMCNT + NOVEC
    IF (CLRKEY .NE. 3) DRMCLR = DRMCNT
    DRMTNS = DRMCLR + NOVEC
    DRMPTR = DRMTNS + NOVEC*2
    DRMPLT = DRMPTD + NOVEC
    TOTDPM = DRMPLT

```

SET00800
SET00810
SET00820
SET00830
SET00840
SET00850
SET00860
SET00870
SET00880
SET00890
SET00900
SET00910
SET00920
SET00930
SET00940
SET00950
SET00960
SET00970
SET00980
SET00990
SET01000
SET01010
SET01020
SET01030
SET01040
SET01050
SET01060
SET01070
SET01080
SET01090
SET01100
SET01110
SET01120
SET01130
SET01140
SET01150
SET01160
SET01170
SET01180
SET01190
SET01200
SET01210
SET01220
SET01230
SET01240
SET01250
SET01260
SET01270
SET01280
SET01290
SET01300
SET01310
SET01320
SET01330
SET01340
SET01350
SET01360
SET01370
SET01380
SET01390
SET01400
SET01410
SET01420
SET01430
SET01440
SET01450
SET01460
SET01470
SET01480
SET01490
SET01500
SET01510
SET01520
SET01530
SET01540
SET01550
SET01560
SET01570
SET01580

ORIGINAL PAGE IS
OF POOR QUALITY

FILF: SETADR

| | |
|--|----------|
| IF (LOG.EQ. 1 .OR. FREQ.EQ. 1) TOTDRM = DRMPLT + 10201 | SET01590 |
| IF ((TOTDRM-DRUMAD) .LE. DRMWDS) GO TO 143 | SET01600 |
| WRITE(A.142)TOTDRM,DRMWDS | SET01610 |
| 142 FORMAT(/' NOT ENOUGH DRUM SPACE.'/ ' TOTAL WORDS OF DRUM SPACE =', | SET01620 |
| • I12/' TOTAL WORDS OF DRUM SPACE AVAILABLE =',I12) | SET01630 |
| CALL CMERR | SET01640 |
| 143 CONTINUE | SET01650 |
| DRMVC1 = DRMVEC | SET01660 |
| DRMID1 = DRMID | SET01670 |
| DRMCR1 = DRMCLR | SET01680 |
| DRMCT1 = DRMCNT | SET01690 |
| DRMTN1 = DRMTNS | SET01700 |
| DRMPT1 = DRMPTR | SET01710 |
| C | SET01720 |
| C | SET01730 |
| C | SET01740 |
| IF (LOG.EQ. 0 .AND. FREQ.EQ. 0) GO TO 145 | SET01750 |
| DO 147 I=1,10201 | SET01760 |
| 147 RUFF(I) = 0 | SET01770 |
| CALL RWRITE(DRMPLT,BUFF,10201,ISTAT) | SET01780 |
| 146 IF (ISTAT.EQ. 1) GO TO 146 | SET01790 |
| 145 RETURN 1 | SET01800 |
| C | SET01810 |
| 150 DEWIND NHSTUN | SET01820 |
| RETURN 2 | SET01830 |
| END | SET01840 |

SET000010
SET000020
SET000030
SET000040
SET000050
SET000060
SET000070
SET000080
SET000090
SET000100
SET000110
SET000120
SET000130
SET000140
SET000150
SET000160
SET000170
SET000180
SET000190
SET000200
SET000210
SET000220
SET000230
SET000240
SET000250
SET000260
SET000270
SET000280
SET000290
SET000300
SET000310
SET000320
SET000330
SET000340
SET000350
SET000360
SET000370
SET000380
SET000390
SET000400
SET000410
SET000420
SET000430
SET000440
SET000450
SET000460
SET000470
SET000480
SET000490
SET000500
SET000510
SET000520
SET000530
SET000540
SET000550
SET000560
SET000570
SET000580
SET000590
SET000600
SET000610
SET000620
SET000630
SET000640
SET000650
SET000660
SET000670
SET000680
SET000690
SET000700
SET000710
SET000720
SET000730
SET000740
SET000750
SET000760
SET000770
SET000780
SET000790

FILE: SET11

```

C      CALL REREAD(30,H0)
C      NOW READ CARD INTO BUFFER
105 READ(21,106) (ACARD(I),I=1,20)
106 FORMAT(20A4)
      WRITE(30,106) (ACARD(I),I=1,20)
      REWIND 30
110 READ(30,110) CODE1,CARD
      FORMAT(A4,6X,62A1)
      REWIND 30
      COL = 0
C
C      WRITE(6,120) CODE1,CARD
120 FORMAT(1X,A4,6X,62A1)
C
      DO 130 I=1,NPOT
      IF (CODE1.EQ. CODE(I)) GO TO (150,180,210,250,290,300,340,
* 350,370,390,400,410,415,420,455,460,470,477),I
130 CONTINUE
      WRITE(6,140)
140 FORMAT(' INVALID CONTROL CARD -- IGNORED')
      GO TO 105
C
C      CHANNEL CARD -- NEEDED ONLY IF STATS FILE IS INPUT
C
150 M=NXTCHR(CARD,COL)
      IF (M.NE. BLANK) GO TO 160
153 WRITE(6,155)
155 FORMAT(' ERROR ON CHANNELS CARD')
      GO TO 105
160 COL = COL + 1
      NOFEAT = NUMBER(CARD,COL,FETVEC,NOFEAT)
      CALL ORDER(FETVEC,NOFEAT)
      NC = NOFEAT
      CLPKFY = 1
      GO TO 105
C
C      STAT FILE CARD
C
180 M = NXTCHR(CARD,COL)
      IF (M.EQ. BLANK) GO TO 105
      IF (M.EQ. URCD) GO TO 190
      IF (M.EQ. FRCD) GO TO 200
185 WRITE(6,187)
187 FORMAT(' ERROR ON STAT FILE CARD')
      GO TO 105
190 J = FIND12(CARD,COL,EQUOM)
      IF (J.NE. 2) GO TO 185
      M = NUMBER(CARD,COL,SAVTAP,ZERO)
      COL = COL + 1
      CLPKFY = 1
      GO TO 180
200 J = FIND12(CARD,COL,EQUOM)
      IF (J.NE. 2) GO TO 185
      M = NUMBER(CARD,COL,STAFIL,ZERO)
      COL = COL + 1
      GO TO 180
C
C      N-DIM HISTOGRAM FILE
C
210 M = NXTCHR(CARD,COL)
      IF (M.EQ. BLANK) GO TO 105
      IF (M.EQ. URCD) GO TO 230
      IF (M.EQ. FRCD) GO TO 240
220 WRITE(6,225)
225 FORMAT(' ERROR ON N-DIM HISTOGRAM FILE CARD')
      GO TO 105
230 J = FIND12(CARD,COL,EQUOM)
      IF (J.NE. 2) GO TO 220
      M = NUMBER(CARD,COL,NHSTUN,ZERO)
      COL = COL + 1
      GO TO 210
240 J = FIND12(CARD,COL,EQUOM)
      IF (J.NE. 2) GO TO 220
      M = NUMBER(CARD,COL,NHSTFI,ZERO)
      COL = COL + 1
      GO TO 210
C
C      PIXEL FREQ. PLOT CARD

```

SET00800
 SET00810
 SET00820
 SET00830
 SET00840
 SET00850
 SET00860
 SET00870
 SET00880
 SET00890
 SET00900
 SET00910
 SET00920
 SET00930
 SET00940
 SET00950
 SET00960
 SET00970
 SET00980
 SET00990
 SET01000
 SET01010
 SET01020
 SET01030
 SET01040
 SET01050
 SET01060
 SET01070
 SET01080
 SET01090
 SET01100
 SET01110
 SET01120
 SET01130
 SET01140
 SET01150
 SET01160
 SET01170
 SET01180
 SET01190
 SET01200
 SET01210
 SET01220
 SET01230
 SET01240
 SET01250
 SET01260
 SET01270
 SET01280
 SET01290
 SET01300
 SET01310
 SET01320
 SET01330
 SET01340
 SET01350
 SET01360
 SET01370
 SET01380
 SET01390
 SET01400
 SET01410
 SET01420
 SET01430
 SET01440
 SET01450
 SET01460
 SET01470
 SET01480
 SET01490
 SET01500
 SET01510
 SET01520
 SET01530
 SET01540
 SET01550
 SET01560
 SET01570
 SET01580

FILE: SET11

```

C
250 M = NXTCHR(CARD,COL)
    IF (M.EQ. BLANK) GO TO 105
    IF (M.EQ. LRCD) GO TO 270
    IF (M.EQ. PRCD) GO TO 280
    IF (M.EQ. RHCD) GO TO 285
255 WRITE(6,250)
260 FORMAT(' ERROR ON OPTION CARD')
    GO TO 105
270 LOG = 1
    GO TO 287
280 FRFO = 1
    GO TO 287
285 XYSCLF = 1
287 J = FIND12(CARD,COL,EQUCOM)
    IF (J.EQ. 3) GO TO 250
    GO TO 105

C
    COLOR CODES
290 NCLSTR = VECSCN(MEANS(ADR+1),NOCHAN,CARD,COL) + NCLSTR
    ADR = NCLSTR*NOCHAN
    NC = NOCHAN
    NOSIIR2 = NCLSTR
    CLPKEY = 2
    GO TO 105

C
    TAPE SIZE CARD
300 M = NXTCHR(CARD,COL)
    IF (M.EQ. BLANK) GO TO 105
    IF (M.EQ. XRCD) GO TO 320
    IF (M.EQ. YRCD) GO TO 330
310 WRITE(315)
315 FORMAT(' ERROR ON TAPE SIZE CARD')
    GO TO 105
320 M = NXTCHR(CARD,COL)
    J = FIND12(CARD,COL,EQUCOM)
    IF (M.EQ. SHCD) CSCALE = 1
    IF (J.NE. 2) GO TO 310
    MM = NUMBER(CARD,COL,J,ZERO)
    IF (M.EQ. HXCD) XHI = J
    IF (M.EQ. LXCD) XLO = J
    IF (M.EQ. SHCD) XSIZ = J
    COL = COL - 1
    GO TO 300
330 M = NXTCHR(CARD,COL)
    IF (M.NE. SHCD) SCSALE = 1
    J = FIND12(CARD,COL,EQUCOM)
    IF (J.NE. 2) GO TO 310
    MM = NUMBER(CARD,COL,J,ZERO)
    IF (M.EQ. HXCD) YHI = J
    IF (M.EQ. LXCD) YLO = J
    IF (M.EQ. SHCD) YSIZ = J
    COL = COL - 1
    GO TO 300

C
    SYMBOLS CARD
340 M = NXTCHR(CARD,COL)
    IF (M.EQ. BLANK) GO TO 105
    IF (M.EQ. KOMMA) GO TO 340
    ICNT = ICNT + 1
    SYMPTX(ICNT) = M
    GO TO 340

C
    MODULE STAT CARD DECK
350 CALL CPDSTA(BUFF, TOP)
    GO TO 105

C
    DATE CARD
370 M = NXTCHR(CARD,COL)
    IF (M.EQ. BLANK) GO TO 105
    READ(30,380) DATE
380 FORMAT(10X,15A4)
    REWIND 30
    GO TO 105

```

ORIGINAL PAGE IS
OF POOR QUALITY

SET01590
SET01600
SET01610
SET01620
SET01630
SET01640
SET01650
SET01660
SET01670
SET01680
SET01690
SET01700
SET01710
SET01720
SET01730
SET01740
SET01750
SET01760
SET01770
SET01780
SET01790
SET01800
SET01810
SET01820
SET01830
SET01840
SET01850
SET01860
SET01870
SET01880
SET01890
SET01900
SET01910
SET01920
SET01930
SET01940
SET01950
SET01960
SET01970
SET01980
SET01990
SET02000
SET02010
SET02020
SET02030
SET02040
SET02050
SET02060
SET02070
SET02080
SET02090
SET02100
SET02110
SET02120
SET02130
SET02140
SET02150
SET02160
SET02170
SET02180
SET02190
SET02200
SET02210
SET02220
SET02230
SET02240
SET02250
SET02260
SET02270
SET02280
SET02290
SET02300
SET02310
SET02320
SET02330
SET02340
SET02350
SET02360
SET02370

FILE: SET11

```

C COMMENT CARD
390 M = NXTCHR(CARD,COL)
IF (M.EQ. BLANK) GO TO 105
READ(30,390) COMMENT
REWIND 30
GO TO 105

C
C HED1
400 M = NXTCHR(CARD,COL)
IF (M.EQ. BLANK) GO TO 105
READ(30,390) HED1
REWIND 30
GO TO 105

C
C HED2
410 M = NXTCHR(CARD,COL)
IF (M.EQ. BLANK) GO TO 105
READ(30,390) HED2
REWIND 30
GO TO 105

C
C SCATTER PLOT TAPE CARD
415 M = NXTCHR(CARD,COL)
IF (M.EQ. BLANK) GO TO 105
IF (M.EQ. UNCD) GO TO 418
416 WRITE(6,417)
417 FORMAT(' ERROR ON SCATTER PLOT TAPE CARD')
GO TO 105
418 J = FIND12(CARD,COL,EQUOM)
IF (J.EQ. 0) GO TO 419
M = NUMBER(CARD,COL,SCRUN,ZERO)
GO TO 105

C
C B - MATRIX CARD
RMKEY = 1 FOR CARDS
RMKEY = 2 FOR FILE
420 M = NXTCHR(CARD,COL)
IF (M.EQ. BLANK) GO TO 433
IF (M.EQ. CHCD) GO TO 450
IF (M.EQ. FHCD) GO TO 450
433 WRITE(6,435)
435 FORMAT(' ERROR ON B-MATRIX CARD')
GO TO 105
440 RMKEY = 1
CALL BMFIL(BMATRX,BMCOMB,BMFEAT,BMVEC,BMKEY)
GO TO 105
450 RMKEY = 2
CALL BMFIL(BMATRX,BMCOMB,BMFEAT,BMVEC,BMKEY)
GO TO 105

C
C BACKGROUND COLOR CARD
455 M = NXTCHR(CARD,COL)
IF (M.EQ. BLANK) GO TO 105
IF (M.EQ. BHCD) BCKGND = 0
IF (M.EQ. WHCD) BCKGND = 255
GO TO 105

C
C R VECTOR
460 NBVCHN = FLOOR(M(CARD,COL,BVEC(NBVCHN+1),30) * NBVCHN)
GO TO 105

C
C SCALING CARD
470 M = NXTCHR(CARD,COL)
IF (M.EQ. BLANK) GO TO 105
IF (M.EQ. FHCD) GO TO 471
IF (M.EQ. YHCD) GO TO 472
IF (M.EQ. XHCD) GO TO 473
IF (M.EQ. WHCD) GO TO 476
GO TO 474
471 SCALRY = 2
GO TO 105
4711 J = FIND12(CARD,COL,EQUOM)

```

SET02380
 SET02390
 SET02400
 SET02410
 SET02420
 SET02430
 SET02440
 SET02450
 SET02460
 SET02470
 SET02480
 SET02490
 SET02500
 SET02510
 SET02520
 SET02530
 SET02540
 SET02550
 SET02560
 SET02570
 SET02580
 SET02590
 SET02600
 SET02610
 SET02620
 SET02630
 SET02640
 SET02650
 SET02660
 SET02670
 SET02680
 SET02690
 SET02700
 SET02710
 SET02720
 SET02730
 SET02740
 SET02750
 SET02760
 SET02770
 SET02780
 SET02790
 SET02800
 SET02810
 SET02820
 SET02830
 SET02840
 SET02850
 SET02860
 SET02870
 SET02880
 SET02890
 SET02900
 SET02910
 SET02920
 SET02930
 SET02940
 SET02950
 SET02960
 SET02970
 SET02980
 SET02990
 SET03000
 SET03010
 SET03020
 SET03030
 SET03040
 SET03050
 SET03060
 SET03070
 SET03080
 SET03090
 SET03100
 SET03110
 SET03120
 SET03130
 SET03140
 SET03150
 SET03160

FILE: SET11

```

IF (J .EQ. 3) GO TO 470
DO TO 105
472 M = NITCHR(CARD,COL)
M = NITCHR(CARD,COL)
J = FIND12(CARD,COL,EQUOM)
IF (J .NE. 2) GO TO 474
J = FLTNUM(CARD,COL,NUM,1)
IF (M .EQ. TRCD) YMIN = NUM
IF (M .EQ. ABCD) YMAX = NUM
SCALY = 1
GO TO 470
473 M = NITCHR(CARD,COL)
M = NITCHR(CARD,COL)
J = FIND12(CARD,COL,EQUOM)
IF (J .NE. 2) GO TO 474
J = FLTNUM(CARD,COL,NUM,1)
IF (M .EQ. TRCD) XMIN = NUM
IF (M .EQ. ABCD) XMAX = NUM
SCALX = 1
GO TO 470
474 WRITE(6,475)
475 FORMAT(' ERROR ON SCALING CARD')
DO TO 105
476 RSCALE = 1
GO TO 471

*END*

477 CONTINUE

480 IF (BMKEY .LE. 0) GO TO 490
IF (LOG .EQ. 1) GO TO 481
IF (FRFQ .EQ. 0) GO TO 482
481 IF (RSCALE .EQ. 1) GO TO 482
WRITE(6,480)
480 FORMAT(' DATA MUST BE RESCALED BEFORE PIXEL FREQUENCY PLOT OPTION
* MAY BE SELECTED')
LOG = 0
FRFQ = 0
482 CONTINUE
CALL W2BM(RMATRIX,BMCOMB,BMFEAT,BMVEC)
GO TO 490
483 WRITE(6,485)NOFET2,BMFEAT
485 FORMAT(' NO. OF PLOTTING CHANNELS, NO. OF B-MATRIX CHANNELS:
* , , RESPECTIVELY')
CALL CMERR

490 CONTINUE

CHANGE MEANS TO FLOATING PT NO
IF (INCLSTR .EQ. 0) GO TO 497
DO 495 I=1,ADR
495 MEANS(I) = MEANS(I)
497 CONTINUE

COMPUTE NO. OF BINS
IF (ICNT .GT. 0) NBINS = ICNT/2
READ HEADER REC FROM N-BIN HIST FILE
READ(NHSTUN) TOTMNS,SIZE,NOFET2,(FETVC2(I),I=1,NOFET2),
NCLRCH,(CLRVEC(I),I=1,NCLRCH)

DEFAULT STAT CHANNELS
IF (CLPKEY .NE. 1) GO TO 499
IF (NOFFAT .NE. 0) GO TO 499
DO 499 I=1,NOFET2
IF (I .GT. 4) GO TO 499
NOFEAT = I
NC = I
499 FETVC(I) = FETVC2(I)
CONTINUE
IF (BMKEY .NE. 0 .AND. NOFET2 .NE. BMFEAT) GO TO 483

```

ORIGINAL PAGE IS
OF POOR QUALITY

SET03178
SET03180
SET03190
SET03208
SET03210
SET03230
SET03240
SET03250
SET03260
SET03270
SET03280
SET03290
SET03300
SET03310
SET03320
SET03330
SET03340
SET03350
SET03360
SET03370
SET03380
SET03390
SET03400
SET03410
SET03420
SET03430
SET03440
SET03450
SET03460
SET03470
SET03480
SET03490
SET03500
SET03510
SET03520
SET03530
SET03540
SET03550
SET03560
SET03570
SET03580
SET03590
SET03600
SET03610
SET03620
SET03630
SET03640
SET03650
SET03660
SET03670
SET03680
SET03690
SET03700
SET03710
SET03720
SET03730
SET03740
SET03750
SET03760
SET03770
SET03780
SET03790
SET03800
SET03810
SET03820
SET03830
SET03840
SET03850
SET03860
SET03870
SET03880
SET03890
SET03900
SET03910
SET03920
SET03930
SET03940
SET03950

FILE: SET11

C ARE COLOR CODES COMING FROM TAPE

C IF (NCLRCH .EQ. 0) GO TO 505
 CLKEY = 3
 NC = NCLRCH
 505 IF (TOTMNS .EQ. 0) GO TO 500
 CLKEY = 4
 NC = NOFET2
 500 CONTINUE

C POSITION TO DESIRED FILE

C CALL FSRSL(NHSTUN,NHSTF1,ISTAT)
 IF (ISTAT .EQ. 0) GO TO 520
 WRITE(6,510)NHSTF1,ISTAT
 510 FORMAT(' ERROR IN POSITIONING N-DIM HIST FILE TO FILE',
 '15/1 ISTAT = ',I5)
 REWIND NHSTUN
 CALL CHERR
 520 CONTINUE

C WRITE(4,700)
 IF (FRFQ .EQ. 1) WRITE(6,710)
 IF (LOG .EQ. 1) WRITE(6,720)
 IF (BMKEY .GT. 0) WRITE(6,740)
 IF (PCKGND .EQ. 0) WRITE(6,750)
 IF (PCKGND .EQ. 255) WRITE(6,760)
 IF (RSCALF .EQ. 0 .AND. BMKEY .GT. 0) WRITE(6,770)
 IF (SCALFY .EQ. 2 .AND. BMKEY .GT. 0) WRITE(6,780)
 IF (SCALFY .EQ. 1) WRITE(6,790)
 IF (CLRKEY .EQ. 1) WRITE(6,800) (FETVEC(I), I=1,NOFET)
 IF (CLRKEY .EQ. 2) WRITE(6,810)
 IF (CLRKEY .EQ. 3) WRITE(6,820)
 IF (CLRKEY .EQ. 4) WRITE(6,830)

700 FORMAT('/// USER HAS SELECTED THE FOLLOWING OPTIONS :',//)
 710 FORMAT(' LINE PRINTER PIXEL FREQ. PLOT')
 720 FORMAT(' LINE PRINTER LOG OF PIXEL FREQ. PLOT')
 740 FORMAT(' TRANSFORM DATA')
 750 FORMAT(' HACK GROUND COLOR FOR SPECTRAL PLOTS WILL BE BLACK')
 760 FORMAT(' HACK GROUND COLOR FOR SPECTRAL PLOTS WILL BE WHITE')
 770 FORMAT(' TRANSFORMED DATA WILL NOT BE RESCALED')
 780 FORMAT(' USER HAS INPUT RESCALING FACTORS')
 790 FORMAT(' RESCALING FACTORS WILL BE COMPUTED FROM N-DIM HIST FILE')
 800 FORMAT(' COLOR CODES ARE COMING FROM STAT FILE USING CHANNELS ',
 '4(IX,I2)')
 810 FORMAT(' COLOR CODES ARE COMING FROM USER INPUT')
 820 FORMAT(' COLOR CODES WILL BE COMPUTED FROM FIELD MEANS')
 830 FORMAT(' COLOR CODES ARE COMING FROM RADIANCE VALUES STORED ON N-
 'DIM HIST FILE')
 RETURN

C END

SET03960
 SET03970
 SET03980
 SET03990
 SET04000
 SET04010
 SET04020
 SET04030
 SET04040
 SET04050
 SET04060
 SET04070
 SET04080
 SET04090
 SET04100
 SET04110
 SET04120
 SET04130
 SET04140
 SET04150
 SET04160
 SET04170
 SET04180
 SET04190
 SET04200
 SET04210
 SET04220
 SET04230
 SET04240
 SET04250
 SET04260
 SET04270
 SET04280
 SET04290
 SET04300
 SET04310
 SET04320
 SET04330
 SET04340
 SET04350
 SET04360
 SET04370
 SET04380
 SET04390
 SET04400
 SET04410
 SET04420
 SET04430
 SET04440
 SET04450
 SET04460
 SET04470

FILE: SORTVC

```

C*      SUBROUTINE SORTVC(HIST,PNTR,ICOL,NOVEC,IRG,IEN,II)
C*      SORTVC SORTS THE ARRAY HIST INTO DESCENDING ORDER
C*      HIST - THE ARRAY TO BE ORDERED
C*      PNTR - ARRAY CONTAINING POINTERS
C*      ICOL - THE COLUMN WITHIN HIST THAT IS TO BE ORDERED
C*      NOVEC - NO. OF VECTORS TO SORT
C*      IRG - ARRAY CONTAINING BEGINNING POINTER FOR EACH BLOCK SORTED
C*      IEN - ARRAY CONTAINING ENDING POINTER FOR EACH BLOCK SORTED
C*      II - NO. OF BLOCK BEING SORTED
C*
C      IMPLICIT INTEGER (A-Z)
C      REAL HIST(ICOL:NOVEC),NUMBR
C      DIMENSION PNTR(1)
C      DIMENSION IRG(1),IEN(1)
C
C      IB = 1
C      IE = 1
C      NEXT = IR
C      DO 200 J=1,NOVEC
C
C      NUMBR = HIST(ICOL,J)
C      COMPARE AGAINST LARGEST NUMBER
C      IF (NUMBR .GE. HIST(ICOL,IB)) GO TO 100
C      COMPARE AGAINST SMALLEST NUMBER
C      IF (NUMBR .LE. HIST(ICOL,IE)) GO TO 120
C      COMPARE AGAINST NEXT LARGEST NUMBER
C      INTRY = PNTR(NEXT)
C      NOVEC1 = J - 1
C      DO 80 I=1,NOVEC1
C      PAST = NEXT
C      NEXT = PNTR(NEXT)
C      IF (NUMBR .GE. HIST(ICOL,NEXT)) GO TO 130
C      IF (NUMBR .GT. HIST(ICOL,INTRY)) GO TO 130
C      PAST = INTRY
C      INTRY = PNTR(INTRY)
C      80 CONTINUE
C      WRITE(6,90)
C      90 FORMAT(' ERROR IN SORTING VECTORS')
C      STOP
C
C      LARGEST NUMBER FOUND THUS FAR
C      100 PNTR(J) = IB
C      IB = J
C      NEXT = IR
C      GO TO 200
C
C      SMALLEST NUMBER FOUND THUS FAR
C      120 PNTR(IE) = J
C      IE = J
C      GO TO 200
C
C      130 PNTR(J) = NEXT
C      NEXT = IR
C      PNTR(PAST) = J
C      200 CONTINUE
C
C      IRG(II) = IR
C      IEN(II) = IE
C      RETURN
C
C      END

```

ORIGINAL PAGE IS
OF POOR QUALITY

FILE: STOFIL

```

C*      SUBROUTINE STOFIL(LIMIT,MEANS,BUFF)
C*      STOFIL READS AND STORES NDIM FILE ON DRUM
C*      IMPLICIT INTEGER (A-Z)
C*      INCLUDE CMRK12.LIST
C*      INCLUDE COMHKA.LIST
C*      COMMON/GLOBAL/HEAD(63),MAPTAP,DATAPE,SAVTAP,RMFILE,BMKEY,
C*      *      HISFIL,HISKEY,TRFORM,ERIPTP,ERPKEY,MAPUNT,NOFILE,
C*      *      DRUMAD,DRUMDS,PAGSIZ,DATAFIL,STAFIL,ASAV,ASAVFL
C*      *      ,NHSTUN,NHSTFI,SCTRUN,MAFFIL
C*      *      ,DOTUNT,DOTFIL,NCHPAS,TRNSFL,RMTRFL,HISTFL,PCHUNT,
C*      *      CRDUNT,PWTUNT,RANDIO
C*      *      COMMON/SCATTER/RSCL,XYSCALE,CLRVEC(30),NCLRCH,CLRKEY,LOG,
C*      *      FREQ,XMAX,YMAX,XMIN,YMIN,ACKGND,XHI,XLO,YLO,XSIZ,
C*      *      YHI,YSIZ,NRINS,SYMTEX(32),RMATX(60),RVEC(30),NBVCHN,NOFEAT
C*      *      ,SCALKY,MFNADR,FLDADR,PNTADR,IDADR,NC,RMFEAT,BMCOMP
C*      *      ,NOVEC,TOTMNS,SIZF,DRMID,DRMID1,DRMCLR,DRMCRI,DRMTNS,DRMTN1,
C*      *      DRMCNT,DRMCT1,DRMVEC,DRMVC1,VECTR1,DATA1,NVEC,NOREAD,LREAD
C*      *      ,DRMPTR,DRMPT1,FETVEC(16),DRMPLT,CSCALE
C*      *      ,NOSUB
CSEND
C      DIMENSION BUFF(1),ISTAT(4,2),ISTA(3,2),MEANS(1)
C      DIMENSION REMDR(4)
C
C      READ MEANS INTO CORE
C      IF (CLRKEY .EQ. 4) READ(NHSTUN) (MEANS(I),I=1,TOTMNS)
C      AMT = 0
C      TOTAL = NOVEC*SIZE
C      DO 100 I=1,4
C      REMD = TOTAL - LIMIT*I
C      IF (REMD .GT. 0) GO TO 90
C      REMDR(I) = TOTAL - LIMIT*(I-1)
C      NTPRDS = 1
C      GO TO 105
C 90  REMDR(I) = LIMIT
C 100 CONTINUE
C 105 CONTINUE
C
C      READ IN REC 4 -- DATA VECTORS
C      DO 130 I=1,NTPRDS
C      NOWRDS = REMDR(I)
C      READ (NHSTUN) (BUFF(J),J=1,NOWRDS)
C
C      DUMP ON HIGH SPEED FRUM
C      CALL RWRITE(DRMVC1,BUFF(1),NOWRDS,ISTAT(I,1))
C      DRMVC1 = DRMVC1 + NOWRDS
C 130 CONTINUE
C      IF (CLRKEY .EQ. 3) NOREC = 3
C      IF (CLRKEY .NE. 3) NOREC = 2
C
C      READ IN REC 5 -- ID ARRAY
C      READ IN REC 6 -- COUNTERS
C      READ IN REC 7 -- COLOR CODES
C
C      DO 200 K=1,NOREC
C      NOWRDS = NOVEC
C      READ(NHSTUN) (BUFF(J),J=1,NOWRDS)
C
C      DUMP ON HIGH SPEED DRUM
C      IF (K .EQ. 1) GO TO 170
C      IF (K .EQ. 2) GO TO 180
C      IF (K .EQ. 3) GO TO 190
C 170 CALL RWRITE(DRMID1,BUFF(1),NOWRDS,ISTA(K,1))
C      DRMID1 = DRMID1 + NOWRDS
C      GO TO 200
C 180 CALL RWRITE(DRMCT1,BUFF(1),NOWRDS,ISTA(K,1))
C      DRMCT1 = DRMCT1 + NOWRDS
C      GO TO 200
C 190 CALL RWRITE(DRMCR1,BUFF(1),NOWRDS,ISTA(K,1))
C      DRMCR1 = DRMCR1 + NOWRDS
C 200 CONTINUE

```

ST000010
ST000020
ST000030
ST000040
ST000050
ST000060
ST000070
ST000080
COM00010
COM00020
COM00030
COM00040
COM00050
COM00060

ST000100
ST000110
ST000120
ST000130
ST000140
ST000150
ST000160
ST000170
ST000180
ST000190
ST000200
ST000210
ST000220
ST000230
ST000240
ST000250
ST000260
ST000270
ST000280
ST000290
ST000300
ST000310
ST000320
ST000330
ST000340
ST000350
ST000360
ST000370
ST000380
ST000390
ST000400
ST000410
ST000420
ST000430
ST000440
ST000450
ST000460
ST000470
ST000480
ST000490
ST000500
ST000510
ST000520
ST000530
ST000540
ST000550
ST000560
ST000570
ST000580
ST000590
ST000600
ST000610
ST000620
ST000630
ST000640
ST000650
ST000660

FILE: STOFIL

CALL FSBSFL (NMSTUN,1,ISTAT1)
RETURN
END

ST000670
ST000680
ST000690

ORIGINAL PAGE IS
OF POOR QUALITY

FILE: TNSFER

```

C      SURROUTINE TNSFER(PLOT,TNSDAT,NVECT,IAAA)
C      TNSFER TRANSFORMS 1 - 16 CHANNELS TO 2 COMPONENTS
C      IMPLICIT INTEGER (A-Z)
C      REAL XMAX,YMAX,XMIN,YMIN
C      REAL TNSDAT
C      INCLUDE COMAK1.LIST
C      INCLUDE COMAK6.LIST
C      INCLUDE COMAK12.LIST
C      COMMON/INFORM/NOCLS2,NOSUR2,NOFET2,VARSZ2,TOTVT2,NOFLO2,
C      *      AVAR2,COVAR2,CLSID2,SUBNO2,SURDS2,FLDSV2,VERTX2,
C      *      FETVC2(30),SURVC2(75),SUHPT(75),CLSV2(60),
C      *      KEPPTS(60),NOGRP,GRPNAM(60),GRPDEX(61),
C      *      GRPCHK(61),GROUPS(124)
C      COMMON/GLOBAL/HEAD(63),MAPTAP,DATE,SAVTAP,BMFILE,BMKEY,
C      *      HISFIL,HISKEY,TRFORM,ERIPTR,ERPKEY,MAPUNT,NOFILE,
C      *      DRUMAD,DRMADS,PAGSIZ,DATFIL,STAFIL,ASAV,ASAVFL
C      *      ,NHSTUN,NHSTFI,SCTRUN,MAPFIL
C      *      ,DOTUNT,DOTFIL,NCHPAS,TRNSFL,BMTRFL,HISTFL,PCHUNT,
C      *      CRDUNT,PRUNT,RANDIO
C      COMMON/SCITFP/RSCALE,XSCALE,CLHVEC(30),NCLRCH,CLRKEY,LOG,
C      *      FREQ,XMAX,YMAX,XMIN,YMIN,BCKGND,XHI,XLO,YLO,XSIZ,
C      *      YHI,YSIZ,NRINS,SYMTX(32),RMATX(60),RVEC(30),NHVCHN,NOFEAT
C      *      ,SCALKY,MFNADR,FLDADR,PNTADR,IDADR,NC,BMFEAT,BMCOMB
C      *      ,NOVEC,TOTMNS,SIZF,DRMID,DRMID1,DRMCLN,DRMCR1,DRMTNS,DRMTN1,
C      *      DRMCNT,DRMCT1,DRMVEC,DRMVC1,VECTR1,DATA1,NVEC,NOREAD,LREAD
C      *      ,DRMPT,DRMPT1,FETVEC(16),DRMPLT,CSCALE
C      *      ,NOSUB
CSEND  DIMENSION DATA(16),PLOT(SIZE,NVECT),TNSDAT(2,NVECT)
C
C      LOGICAL*1 LDUM(4),LLDUM(4)
C      EQUIVALENCE (IDUM,LDUM(1)),(IIDUM,LLDUM(1))
C      DO 100 I=1,NVECT
C      DO 10 II=1,NOFET2
C      III=(II-1)/4+1
C      IRYTF=II-((II-1)/4)*4
C      IDUM=PLOT(III,I)
C      IIDUM=0
C      LLDUM(4)=LDUM(1*BYTE)
C 10  DATA(II)=IIDUM
C
C      TRANSFORM DATA VECTOR
C      CALL MATTNS(RMATX,DATA,TNSDAT(1,I),RVEC,BMCOMB,BMFEAT)
C
C      PSCALE = 1  USER HAS INPUT SCALE FACTORS
C      RSCALE = 2  COMPUTE SCALE FACTORS
C
C      USER HAS INPUT SCALE FACTORS
C      IF (SCALKY.EQ. 2) GO TO 20
C      SWITCH = 1
C      IF (RSCALE.EQ. 1) CALL RESCLE(TNSDAT(1,I),SWITCH,NVECT)
C      GO TO 100
C
C      YMAX, XMAX, AND XMIN ARE TO BE TAKEN FROM HIST FILE
C 20  IF ( I.NE. 1) GO TO 25
C      IF ( II.NE. 1) GO TO 25
C      XMIN = TNSDAT(1,I)
C      YMIN = TNSDAT(2,I)
C 25  IF (XMAX.LT. TNSDAT(1,I)) XMAX = TNSDAT(1,I)
C      IF (XMIN.GT. TNSDAT(1,I)) XMIN = TNSDAT(1,I)
C      IF (YMAX.LT. TNSDAT(2,I)) YMAX = TNSDAT(2,I)
C      IF (YMIN.GT. TNSDAT(2,I)) YMIN = TNSDAT(2,I)
C 100 CONTINUE
C
C      RETURN
C      END

```

FILE: UNPCKV

```

C      SURROUTINE UNPCKV(PLOT,TNSDAT,NVECT)
C
C      UNPCKV UNPACKS TWO 8 BITS BYTES AND STORES THE VALUES INTO TWO
C      FLOATING POINT WORDS
C      IMPLICIT INTEGER (A-Z)
C      REAL TNSDAT(2,NVECT)
C
C      INCLUDE CMH12.LIST
COMMON/SCITED/NSCALE,XYSCLF,CLRVEC(30),NCLRCH,CLRKEY,LOG,
* FREQ,XMAX,YMAX,XMIN,YMIN,RCKGND,XHI,XLO,YLO,XSIZ,
* YHI,YSIZ,NRINS,SYMMTX(32),RMATRX(60),HVEC(30),NRVCHN,NOFEAT
* SCALKY,MENADP,FLDADW,PNTADP,IDADP,NC,BMFEAT,BMCOMR
* NOVEC,TOTYNS,SIZE,DRMID,DRMID1,DRMCLR,DRMCRI,DRMTNS,DRMTN1,
* DRMCNT,DRVCT1,DRMVEC,DRMVC1,VECTR1,DATA1,NVEC,NOREAD,LREAD
* DRMPTR,DRMPT1,FETVEC(16),DRMPLT,CSCALE
* NOSUB
C$END
C      DIMENSION PLOT(1)
LOGICAL*1 LDUM(4),LLDUM(4)
EQUIVALENCE (LDUM(1),IDUM),(LLDUM(1),IIDUM)
C
DO 100 I=1,NVECT
  IDUM=PLOT(I)
  IIDUM=0
  LLDUM(4)=LDUM(2)
  VALUE2=IIDUM
  LLDUM(4)=LDUM(1)
  VALUE1=IIDUM
  TNSDAT(2,I) = VALUE2
100 TNSDAT(1,I) = VALUE1
  RETURN
END
```

UNP00010
UNP00020
UNP00030
UNP00040
UNP00050
UNP00060
UNP00070
UNP00080
UNP00090
UNP00100
UNP00110
UNP00120
UNP00130
UNP00140
UNP00150
UNP00160
UNP00170
UNP00180
UNP00190
UNP00200
UNP00210
UNP00220
UNP00230
UNP00240
UNP00250
UNP00260
UNP00270
UNP00280
UNP00290
UNP00300
UNP00310
UNP00320
UNP00330
UNP00340
UNP00350
UNP00360

ORIGINAL PAGE IS
OF POOR QUALITY

FILE: VECSCN

| | | |
|---|---|----------|
| C | FUNCTION VECSCN(VECTR,NVCELT,CARD,COL) | VEC00010 |
| C | VECSCN CONVERTS ALPHA CHARACTERS TO INTEGERS | VEC00020 |
| C | RETURNS THE NO. OF ELEMENTS WITHIN A SET OF PARENTHESES | VEC00030 |
| C | RETURNS THE NO. OF PARENTHESES | VEC00040 |
| C | IMPLICIT INTEGER (A-Z) | VEC00050 |
| C | DIMENSION VECTR(1),COMMA(2),CARD(1) | VEC00060 |
| C | DATA STAR /'/' ,BLANK/' ' ,LPBCD/'(' ,RPRCD/')' / | VEC00070 |
| C | DATA COMMA/' , ' ,KOMMA/' , ' / | VEC00080 |
| C | NTIMES = 0 | VEC00090 |
| C | TOTNUM = 0 | VEC00100 |
| C | 80 TOTNUM = TOTNUM + 1 | VEC00110 |
| C | NVCELT = NVCELT + 1 | VEC00120 |
| C | 100 M = NATCHR(CARD,COL) | VEC00130 |
| C | IF (M.EQ. BLANK) GO TO 140 | VEC00140 |
| C | IF (M.EQ. LPBCD) GO TO 100 | VEC00150 |
| C | IF (M.EQ. RPRCD) GO TO 130 | VEC00160 |
| C | IF (M.EQ. KOMMA) GO TO 120 | VEC00170 |
| C | IF (M.EQ. STAR) GO TO 135 | VEC00180 |
| C | CHANGING NUMBER FROM ALPHA MODE TO INTEGER MODE | VEC00190 |
| C | 110 CALL I4A1RN(CARD(COL),1,NUM) | VEC00200 |
| C | NUMB = 10*NUMB + NUM | VEC00210 |
| C | IF (NUM.LT. 0 .OR. NUM.GT. 9) GO TO 150 | VEC00220 |
| C | GO TO 100 | VEC00230 |
| C | FOUND A COMMA | VEC00240 |
| C | 120 VECTR(TOTNUM) = NUMB | VEC00250 |
| C | NUMB = 0 | VEC00260 |
| C | GO TO 80 | VEC00270 |
| C | FOUND A '(' | VEC00280 |
| C | 130 VECTR(TOTNUM) = NUMB | VEC00290 |
| C | NUMB = 0 | VEC00300 |
| C | IF (NTIMES.EQ. 0) GO TO 133 | VEC00310 |
| C | DO 132 I=1,NTIMES | VEC00320 |
| C | DO 132 J=1,NVCELT | VEC00330 |
| C | JJ = TOTNUM + (I-1)*NVCELT + J | VEC00340 |
| C | 132 VECTR(JJ) = VECTR(TOTNUM-NVCELT+J) | VEC00350 |
| C | TOTNUM = TOTNUM + NTIMES*NVCELT | VEC00360 |
| C | NTIMES = 0 | VEC00370 |
| C | 133 J = FIND12(CARD,COL,COMMA) | VEC00380 |
| C | IF (J.EQ. -1) GO TO 140 | VEC00390 |
| C | NVCELT = 0 | VEC00400 |
| C | GO TO 80 | VEC00410 |
| C | FOUND A MULTIPLICATIVE FACTOR | VEC00420 |
| C | 135 NTIMES = NUMB - 1 | VEC00430 |
| C | NUMB = 0 | VEC00440 |
| C | GO TO 100 | VEC00450 |
| C | FINISHED SCANNING CARD | VEC00460 |
| C | 140 VECSCN = TOTNUM/NVCELT | VEC00470 |
| C | RETURN | VEC00480 |
| C | 150 VECSCN = -1 | VEC00490 |
| C | WRITE(A,200) | VEC00500 |
| C | 200 FORMAT(' ERROR OCCURRED SCANNING VECTOR CARD') | VEC00510 |
| C | RETURN | VEC00520 |
| C | END | VEC00530 |
| | | VEC00540 |
| | | VEC00550 |
| | | VEC00560 |
| | | VEC00570 |
| | | VEC00580 |
| | | VEC00590 |
| | | VEC00600 |
| | | VEC00610 |
| | | VEC00620 |
| | | VEC00630 |
| | | VEC00640 |
| | | VEC00650 |
| | | VEC00660 |
| | | VEC00670 |
| | | VEC00680 |
| | | VEC00690 |
| | | VEC00700 |
| | | VEC00710 |
| | | VEC00720 |
| | | VEC00730 |

17. DOTDATA PROCESSOR

FILE: DOTDAT

```
C      DOTDAT IS THE DRIVER ROUTINE FOR THE DOTDATA PROCESSOR
C
C      SURROUTINE DOTDAT(ARRAY, TOP)
C
      DIMENSION ARRAY(1)
      CALL SFT13
      CALL DOT5(ARRAY(1), ARRAY(5001), ARRAY(6001), TOP)
      RETURN
      END
```

ORIGINAL PAGE IS
OF POOR QUALITY

FILE: DOTS

```

C      DOTS IS THE CO-ORDINATOR FOR CREATING THE DOT DATA FILE
C
C      SUBROUTINE DOTS(DATA,FIELDS,VERTEX,TOP)
C      IMPLICIT INTEGER (A-Z)
C      DIMENSION DATA(SIZE,1),IDATA(10000)
C      DIMENSION FIELDS(4,1),          FL(12),VERTEX(1)
C      INCLUDE COM4K1.LIST
C      INCLUDE COM4K6.LIST
C      INCLUDE COM4K14.LIST
C      INCLUDE COM4K16.LIST
C
C      COMMON/INFORM/NOCLS2,NOSUR2,NOFET2,VARS22,TOTVT2,NOFLD2,
C      *      AVAR2,COVAR2,CLSID2,SURN02,SURDS2,FLDSV2,VERTX2,
C      *      FETVC2(30),SURVC2(75),SURPTH(75),CLVC2(60),
C      *      KEPPTS(60),NOGRP,GRPNAM(60),GRPD(61),
C      *      GRPCHK(61),GROUPS(124)
C      COMMON/GLOBAL/HEAD(63),MAPTAP,DATAPF,SAVTAP,BMFILE,BMKEY,
C      *      HISFIL,HISKEY,TRFORM,FRIPF,ERPKEY,MAPUNT,NOFILE,
C      *      DRUMAD,DRUMDS,PAGSIZ,DATFIL,STAFIL,ASAV,ASAVL
C      *      ,NHSTUN,NHSTFI,STRUN,MAPFIL
C      *      ,DOTUNT,DOTFIL,NCHPAS,TRNSFL,BMTRFL,HISTFL,PCHUNT,
C      *      CRDUNT,PRTUNT,HANDIO
C      COMMON /DOTVEC/TYPE,CATNAM(60),NOCAT,TOTVEC,FLDINF(6),PRTKEY
C      *      ,SIZE,LACIE
C      COMMON/ISOLNK/SUNANG(8),ISUNT,ISUNC,SMSTR,SMSTP,SMINC,LINSKP
CSEND
C
C      DATA BLANK/' '/
C      NOSUN = 8
C      ISUNT = 1
C      ISUNC = 0
C      STAMNT = 1
C      SWITCH = 0
C      IPT = 1
C      IF(LACIE.NE.0)IPT=0
C
C*** CODE ADDED NOV 21, 1978 TO SUPPORT LIST PROCESSING
C
C      INIT = 0
C      ZERO = 0
C      SWCHG = 0
C      NOCAT = 0
C      NOFLD2=0
C      TYPE = 1
C
C      INITIALIZE IMAGE DATA TAPE
C
C 90 CALL TAPHDR(DATAPF,DATFIL)
C
C      DO A0 I=1,SIZE
C      DO A0 J=1,TOTVEC
C  A0 DATA(I,J) = 0
C      TOTVT2 = 0
C      TOTVEC = 0
C  A5 IF(LACIE.EQ.1)CALL FLDLAC(FIELDS,STAMNT,&100,&510,&520,IPT,
C      *VERTEX)
C
C*** ADDED NOV 21, 1978 IN SUPPORT OF LIST PROCESSING
C
C      IF (LACIE.GT.1) CALL LISTLC(FIELDS,STAMNT,&100,&510,
C      *&520,SWCHG,INIT,LACIE,ZERO,IPT,VERTEX)
C      CALL FLDTYP(FIELDS,STAMNT,&100,&510,&520,IPT,VERTEX)
C
C 100 LINSTR = FLDINF(1)
C      LINEND = FLDINF(2)
C      LININC = FLDINF(3)
C      SAMSTR = FLDINF(4)
C      SAMEND = FLDINF(5)
C      SAMINC = FLDINF(6)
C      FIELDS(2,NOFLD2) = NOCAT
C      TOTVT2 = FIELDS(4,NOFLD2) + TOTVT2
C
C      ILINE = (LINEND-LINSTR)/LININC + 1
C      NSAMP = (SAMEND-SAMSTR)/SAMINC + 1
C
C      POSITION IMAGE TAPE FOR THIS FIELD
C      CALL FLDINT(FLDINF(1),FETVC2,NOFET2)

```

DOT00010
 DOT00020
 DOT00030
 DOT00040
 DOT00050
 DOT00060
 DOT00070
 DOT00080
 DOT00090
 DOT00100
 DOT00110
 DOT00120
 DOT00130
 DOT00140
 DOT00150
 DOT00160
 DOT00170
 DOT00180
 DOT00190
 DOT00200
 DOT00210
 DOT00220
 DOT00230
 DOT00240
 DOT00250
 DOT00260
 DOT00270
 DOT00280
 DOT00290
 DOT00300
 DOT00310
 DOT00320
 DOT00330
 DOT00340
 DOT00350
 DOT00360
 DOT00370
 DOT00380
 DOT00390
 DOT00400
 DOT00410
 DOT00420
 DOT00430
 DOT00440
 DOT00450
 DOT00460
 DOT00470
 DOT00480
 DOT00490
 DOT00500
 DOT00510
 DOT00520
 DOT00530
 DOT00540
 DOT00550
 DOT00560
 DOT00570
 DOT00580
 DOT00590
 DOT00600
 DOT00610
 DOT00620
 DOT00630
 DOT00640
 DOT00650
 DOT00660
 DOT00670
 DOT00680
 DOT00690
 DOT00700
 DOT00710
 DOT00720
 DOT00730
 DOT00740
 DOT00750
 DOT00760
 DOT00770
 DOT00780
 DOT00790

FILE: DOTS

```

C      READ A SCAN LINE OF DATA, AND PROCESS IT
C      DO 500 LINE=LINSTR,LINEND,LININC
C      NLINE = NLINE + 1
C      CALL LINEHD(IDATA,ENDTAP)
C      IF (ENDTAP.EQ.-1) GO TO 600
C      FIND INTERSECTIONS FOR N-R FIELDS
C      CALL FDLINT(VERTEX(IPT),FIELDS(4,NOFLD2),FL,LINE,SAMP,NI)
C      DO 400 J=1,NI,2
C      IB = (FL(J)-SAMSTR)/SAMINC + 1
C      IE = (FL(J+1)-SAMSTR)/SAMINC + 1
C      IF (MOD(SAMSTR,SAMINC).NE.MOD(FL(J),SAMINC)) IB = IB + 1
C      IF (IB.GT.IE) GO TO 400
C
C      COLLECTING INFO FOR DATA REC FOR DOTFIL
C      DO 350 K=IB,IE
C      TOTVEC = TOTVEC + 1
C      IF (TOTVEC.LE.250) GO TO 110
C      TOTVEC=250
C      WRITE(6,108)
C      108A FORMAT(7,109)
C      WRITE(6,109)
C      109A FORMAT(25X,'***** NOTE - TOTVEC WAS GREATER THAN 250. THEREFORE TOT
C      1VEC WAS SET TO 250 *****')
C      GO TO 600
C      110 CONTINUE
C      DATA(1,TOTVEC) = SAMSTR + SAMINC * (K-1)
C      DATA(2,TOTVEC) = LINE
C      DATA(3,TOTVEC) = TYPE
C      DATA(4,TOTVEC) = NOCAT
C      DO 150 I=1,NOFET2
C      KK = (I-1)*NSAMP + K
C      150 DATA(4+I,TOTVEC) = IDATA(KK)
C      350 CONTINUE
C      400 CONTINUE
C      500 CONTINUE
C      600 CONTINUE
C      GO TO 85
C
C      WRITE DOT DATA FILE
C      510 CALL WPTFLD(FIELDS,VERTEX,NOFLD2,2,CATNAM,DUMMY)
C      CALL WRDOT(TOTVEC,NOSUN,FIELDS,VERTEX,SUNANG,DATA,NOCAT,
C      * CATNAM,SIZE,NOFET2,FETVC2,TOTVT2,NOFLD2,
C      * DOTUNT,DOTFIL)
C      DOTFIL = DOTFIL + 1
C      NOCAT = 0
C      SWITCH = 1
C      IF (LACIE.NE.0) NOFLD2=0
C      IF (PRTKEY.EQ.1) GO TO 530
C      GO TO 90
C
C      SEND CARD FOUND
C      520 CALL WPTFLD(FIELDS,VERTEX,NOFLD2,2,CATNAM,DUMMY)
C      CALL WRDOT(TOTVEC,NOSUN,FIELDS,VERTEX,SUNANG,DATA,NOCAT,
C      * CATNAM,SIZE,NOFET2,FETVC2,TOTVT2,NOFLD2,
C      * DOTUNT,DOTFIL)
C      SWITCH = 0
C      IF (PRTKEY.EQ.1) GO TO 530
C
C      ROUTINE TO PRINT DOT DATA RECORD
C      530 CONTINUE
C      700 FORMAT(7,109)
C      600 FORMAT(1X,' NO.',2X,'SAMPLF',2X,'LINE',2X,'TYPE',2X,'CATEGORY',

```

DOT00400
 DOT00810
 DOT00820
 DOT00830
 DOT00840
 DOT00850
 DOT00860
 DOT00870
 DOT00880
 DOT00890
 DOT00900
 DOT00910
 DOT00920
 DOT00930
 DOT00940
 DOT00950
 DOT00960
 DOT00970
 DOT00980
 DOT00990
 DOT01000
 DOT01010
 DOT01020
 DOT01030
 DOT01040
 DOT01050
 DOT01060
 DOT01070
 DOT01080
 DOT01090
 DOT01100
 DOT01110
 DOT01120
 DOT01130
 DOT01140
 DOT01150
 DOT01160
 DOT01170
 DOT01180
 DOT01190
 DOT01200
 DOT01210
 DOT01220
 DOT01230
 DOT01240
 DOT01250
 DOT01260
 DOT01270
 DOT01280
 DOT01290
 DOT01300
 DOT01310
 DOT01320
 DOT01330
 DOT01340
 DOT01350
 DOT01360
 DOT01370
 DOT01380
 DOT01390
 DOT01400
 DOT01410
 DOT01420
 DOT01430
 DOT01440
 DOT01450
 DOT01460
 DOT01470
 DOT01480
 DOT01490
 DOT01500
 DOT01510
 DOT01520
 DOT01530
 DOT01540
 DOT01550
 DOT01560
 DOT01570
 DOT01580

FILE: DOTS

| | | |
|-----|---|----------|
| 1 | 30X,'DATA'/) | DOT01590 |
| | ISTART=1 | DOT01600 |
| | IEND=10 | DOT01610 |
| 709 | CONTINUE | DOT01620 |
| | IKT=0 | DOT01630 |
| | DO 800 11=1,TOTVEC | DOT01640 |
| | IKT=IKT+1 | DOT01650 |
| | IF (IEND.GT.NOFEI2) IEND=NOFEI2 | DOT01660 |
| | IF (11.NE.1.AND.IKT.EQ.1) WRITE(6,810) | DOT01670 |
| 810 | FORMAT(1H,5(/)) | DOT01680 |
| | IF (IKT.NE.1) GO TO 820 | DOT01690 |
| | WRITE(6,700) | DOT01700 |
| | WRITE(6,690) | DOT01710 |
| | WRITE(6,720) (BLANK,FETVC2(I),I=ISTART,IEND) | DOT01720 |
| 720 | FORMAT(37X,10(A1,'CH('',12,'')')) | DOT01730 |
| 820 | CONTINUE | DOT01740 |
| | WRITE(6,710) 11,(DATA(1,11),I=1,4),(DATA(4+JJ,11),JJ=ISTART,IEND) | DOT01750 |
| 710 | FORMAT(1X,13,1H,.3X,14,3X,14,2X,12,6X,12,8X,10(13,4X)) | DOT01760 |
| | WRITE(6,712) | DOT01770 |
| 712 | FORMAT() | DOT01780 |
| 800 | CONTINUE | DOT01790 |
| | IF (NOFEI2.GT.10) GO TO 830 | DOT01800 |
| | GO TO 840 | DOT01810 |
| 830 | CONTINUE | DOT01820 |
| | IF (ITWO.EQ.1) GO TO 840 | DOT01830 |
| | ITWO=1 | DOT01840 |
| | ISTART=IEND+ISTART | DOT01850 |
| | IEND=NOFEI2 | DOT01860 |
| | GO TO 709 | DOT01870 |
| 840 | CONTINUE | DOT01880 |
| | IF (SWTCH.EQ.1) GO TO 90 | DOT01890 |
| | RETURN | DOT01900 |
| | END | DOT01910 |

ORIGINAL PAGE IS
OF POOR QUALITY

FILE: FLDLAC

```

C      FIELDS - CATEGORY NAME AND DOT TYPE FOR DOT 1 STORED IN      FLD00010
C      FIELD(1,1) AND FIELD(4,1)                                     FLD00020
C      STAMNT - INITIALLY SET TO 1. SWITCHED TO INDICATE DOTS BEING   FLD00030
C      TAKEN FROM CURRENTLY READ CARD.                                FLD00040
C      IPT - INITIALLY SET TO 1. INDEX NUMBER FOR FIELD VERTEX INFORMATION FLD00050
C      VERTEX - VERTEX INFORMATION FOR EACH DOT.                      FLD00060
C                                                                      FLD00070
C      SUBROUTINE FLDLAC(FIELDS,STAMNT,*,*,*,IPT,VERTEX)             FLD00080
C      IMPLICIT INTEGER (A-Z)                                         FLD00090
C      LOGICAL*1 LCARD(300),LCATNM(4)                                FLD00100
C      REAL DIM                                                        FLD00110
C      DIMENSION FIELDS(4,1),VERTEX(1),CARD(75),NDOTS(30)           FLD00120
C      DIMENSION ACARD(80)                                             FLD00130
C      LOGICAL SWITCH                                                  FLD00140
C      DATA SWITCH/.TRUE./,SWCHG/0/,ENDBCD/'SEN'/,                  FLD00150
C      *CATNM/' '/,                                                  FLD00160
C      INCLUDE CMH14                                                  FLD00170
C      INCLUDE CMH11                                                  FLD00180
C      COMMON/INFORM/NOCL52,NOSUM2,NOFET2,VARSZ2,TOTVT2,NOFLD2,      FLD00190
C      AVAR2,CVAR2,CLS102,SUBN02,SURDS2,FLOS2,VERTX2,               FLD00200
C      FETVC2(30),SURVC2(75),SUMPTR(75),CLSYC2(60),                 FLD00210
C      KFPPTS(60),NOGWP,GWPNAM(60),GWPDEX(61),                      FLD00220
C      GRPCHK(61),GROUPS(124)                                         FLD00230
C      COMMON /DOTVEC/TYPE,CATNAM(40),NOCAT,TOTVEC,FLDINF(6),PRTKEY  FLD00240
C      ,SIZE,LACIE                                                    FLD00250
C                                                                      FLD00260
CSEND  EQUIVALENCE (LCATNM(1),CATNM),(LCARD(1),CARD(1))             FLD00270
C      IF(STAMNT.EQ.2)GO TO 30                                         FLD00280
C      IF(.NOT.SWITCH)GO TO 20                                         FLD00290
C      CALL HFWF41(30,80)                                              FLD00300
C      READ(21,103)(ACARD(I),I=1,80)                                  FLD00310
C      FORMAT(A0A1)                                                    FLD00320
C      WRITE(30,103)(ACARD(I),I=1,80)                                  FLD00330
C      REWIND 30                                                        FLD00340
C      READ(30,1000)ID,TYPE,CARD                                       FLD00350
C      REWIND 30                                                        FLD00360
C      FORMAT(A3,1X,I1,75A1)                                           FLD00370
C      IF(TYPE.EQ.1)GO TO 20                                           FLD00380
C      IF(SWCHG.NE.0)GO TO 40                                           FLD00390
C      TYPE = 1TYPE                                                    FLD00400
C                                                                      FLD00410
C      READ CARD                                                        FLD00420
C                                                                      FLD00430
C      COL = 0                                                         FLD00440
C      CATNM = N*CHW(CARD,COL)                                         FLD00450
C      IF NEXT CHAR IS NOT A CAT. NAME, CORRECT COL COUNT TO READ NUM FLD00460
C      IF(CATNM.GT.0)GO TO 21                                           FLD00470
C      LINDEX=4*COL+1                                                  FLD00480
C      LCATNM(2)=LCARD(LINDEX)                                         FLD00490
C      COL=COL+1                                                        FLD00500
C      IF(CATNM.EQ.CATNM)GO TO 23                                       FLD00510
C      NOCAT=NOCAT+1                                                    FLD00520
C      CATNAM(NOCAT)=CATNM                                              FLD00530
C      CATNM = CATNM                                                    FLD00540
C      GO TO 23                                                         FLD00550
C      COL=COL-1                                                        FLD00560
C      NOCARD=0                                                         FLD00570
C      CALL NUMRD(NDOTS,NDCARD,CARD,COL)                                FLD00580
C      IF(NDCARD.EQ.0)GO TO 10                                           FLD00590
C      ICNT = 0                                                         FLD00600
C      STAMNT = 2                                                         FLD00610
C      SWITCH = .TRUE.                                                  FLD00620
C      GO TO 100                                                        FLD00630
C                                                                      FLD00640
C      TEST FOR END OF DOTS TO BE PROCESSED ON CARD                  FLD00650
C      IF(ICNT.LT.NDCARD)GO TO 100                                       FLD00660
C      READ NEXT CARD                                                  FLD00670
C                                                                      FLD00680
C      STAMNT = 1                                                       FLD00690
C      ICNT = 0                                                         FLD00700
C      READ(21,103)(ACARD(I),I=1,80)                                  FLD00710
C      WRITE(30,103)(ACARD(I),I=1,80)                                  FLD00720
C      REWIND 30                                                        FLD00730
C      READ(30,1000)ID,TYPE,CARD                                       FLD00740
C      REWIND 30                                                        FLD00750
C      IF(ID.EQ.FNDHCD)RETURN 3                                         FLD00760
C      IF(TYPE.EQ.1TYPE)GO TO 20                                         FLD00770
C      IF(TYPE.EQ.1TYPE)GO TO 20                                         FLD00780
C      IF(TYPE.EQ.1TYPE)GO TO 20                                         FLD00790

```

FILE: FLDLAC

```

        SWITCH = .FALSE.
        SWCHG = SWCHG + 1
        IF (SWCHG.GT.1) GO TO 40
        TYPE = TYPES
        IPT = 0
C***** CHANGED JUNE 28 1978
        RETURN 2
C
C 100 ICNT = ICNT + 1
        NOFLD2 = NOFLD2 + 1
C
C      COMPUTE LINE INCREMENT
        NN = NDOTS(ICNT)
        NI = IARS(NN) / 100000000
        LI = IARS(NN) - NI * 100000000
        IF (LI.GE.100000000) NI = NI + 1
C
C      COMPUTE SAMPLE INCREMENT
        KK=1
        IF (NN.LT.0) KK=-1
        LI = NI * KK
        N2 = NN - LI * 100000000
        N3 = IARS(N2) / 10000
        SI = IARS(N2) - N3 * 10000
        IF (SI.GE.1000) N3 = N3 + 1
        KK=1
        IF (N2.LT.0) KK=-1
        SI = N3 * KK
        LACI = N2 - SI * 10000
        LR = (LACI-1) / 19
        LP = (LR+1) * 10
        LS = LR - 1
        LS = LS / 10
        LS = 10 * (LACI - (LS*19))
        L = LR - LI
        S = LS + SI
C
C      STORE DOT INFO
        FIELDS(1,NOFLD2) = CATNM
        FIELDS(4,NOFLD2) = 2
        FLDINF(1) = L
        FLDINF(2) = L
        FLDINF(3) = L
        FLDINF(4) = S
        FLDINF(5) = S
        FLDINF(6) = 1
        IF (IPT.NE.0) GO TO 35
        IPT = -3
C 35 IPT = IPT + 4
        VERTX(IPT) = S
        VERTX(IPT+1) = L
        VERTX(IPT+2) = S
        VERTX(IPT+3) = L
        RETURN 1
C 40 WRITE(4,2000)
2000 FORMAT(/,4X,'ERROR HAS OCCURRED IN READING LACIE FORMATTED DOT CAR
        DS = SUBROUTINE FLULAC - EXIT TAKEN')
        RETURN 3
        END

```

FL000A00
 FL000A10
 FL000A20
 FL000A30
 FL000A40
 FL000A50
 FL000A60
 FL000A70
 FL000A80
 FL000A90
 FL000900
 FL000910
 FL000920
 FL000930
 FL000940
 FL000950
 FL000960
 FL000970
 FL000980
 FL000990
 FL001000
 FL001010
 FL001020
 FL001030
 FL001040
 FL001050
 FL001060
 FL001070
 FL001080
 FL001090
 FL001100
 FL001110
 FL001120
 FL001130
 FL001140
 FL001150
 FL001160
 FL001170
 FL001180
 FL001190
 FL001200
 FL001210
 FL001220
 FL001230
 FL001240
 FL001250
 FL001260
 FL001270
 FL001280
 FL001290
 FL001300
 FL001310
 FL001320
 FL001330
 FL001340
 FL001350
 FL001360
 FL001370
 FL001380
 FL001390
 FL001400
 FL001410
 FL001420

FILE: FLDTYP

```

SUBROUTINE FLDTYP(FIELDS,STAMNT,*,*,*,IPT,VERTEX)
  IMPLICIT INTEGER (A-Z)
  INCLUDE COMRK1.LIST
  INCLUDE COMRK1A.LIST
  COMMON/INFORM/NOCLS2,NOSUR2,NOFET2,VARSZP,TOTVTP,NOFLD2,
  AVAR2,COVAR2,CLS1D2,SUBNO2,SUBD2,FLDSV2,VERTX2,
  FETVC2(30),SURVC2(75),SUBPTR(75),CLSV2(60),
  KFPPTS(60),NOGRP,GRPNAM(40),GRPDEX(41),
  GRPCHR(41),RGROUPS(124)
  COMMON /DOTVEC/TYPE,CATNAM(60),NOCAT,TOTVEC,FLDINF(6),PRTKEY
  *SIZE,LACIE
  DIMENSION FIELDS(4,1),VERTEX(1)
  DIMENSION CARD(62)
  LOGICAL SWITCH
  DATA SWITCH/TRUE/
  IPT = IPT + FIELDS(4,NOFLD2)*2
  GO TO 190,100,STAMNT
80  I = LAREAD(FIELDS(1,NOFLD2+1),VERTEX(IPT),FLDINF(1),FIELDS(4,NOFLD2
  *2,1))
  WAS CLASS,SUBCLASS,FIELD, OR SEND* ENCOUNTERED
  IF (I) :EO: -4) GO TO 90
  IF (I) :EO: -1) GO TO 120
  IF (I) :EO: 1) GO TO 130
  IF (I) :EO: 0) GO TO 140
  TYPE CARD
  90  IF (SWITCH) GO TO 100
  STAMNT = 2
  RETURN 2
100  READ(30,115)CARD
115  FORMAT(10X,62A1)
  REWIND 30
  COL = 0
  ZERO = 0
  M = NUMBER(CARD,COL,TYPE,ZERO)
  IPT = 1
  NOFLD2 = 0
  NOCAT = 0
  SWITCH = .FALSE.
  GO TO 80
  CLASSNAME CARD
110  FORMAT(10X,A4)
120  NOCAT = NOCAT + 1
  READ(30,110)CATNAM(NOCAT)
  REWIND 30
  GO TO 80
  FIELD CARD
130  NOFLD2 = NOFLD2 + 1
  STAMNT = 1
  RETURN 1
  SEND*
140  SWITCH=.TRUE.
  RETURN 3
END

```

FLD000010
 FLD000020
 FLD000030
 FLD000040
 FLD000050
 FLD000060
 FLD000070
 FLD000080
 FLD000090
 FLD000100
 FLD000110
 FLD000120
 FLD000130
 FLD000140
 FLD000150
 FLD000160
 FLD000170
 FLD000180
 FLD000190
 FLD000200
 FLD000210
 FLD000220
 FLD000230
 FLD000240
 FLD000250
 FLD000260
 FLD000270
 FLD000280
 FLD000290
 FLD000300
 FLD000310
 FLD000320
 FLD000330
 FLD000340
 FLD000350
 FLD000360
 FLD000370
 FLD000380
 FLD000390
 FLD000400
 FLD000410
 FLD000420
 FLD000430
 FLD000440
 FLD000450
 FLD000460
 FLD000470
 FLD000480
 FLD000490
 FLD000500
 FLD000510
 FLD000520
 FLD000530
 FLD000540
 FLD000550
 FLD000560
 FLD000570
 FLD000580
 FLD000590
 FLD000600
 FLD000610
 FLD000620
 FLD000630
 FLD000640
 FLD000650
 FLD000660
 FLD000670

ORIGINAL PAGE IS
OF POOR QUALITY

www

CSEND

C

CCC

3

C

3

```

154 J = FIND12(CARD,COL,EDUCOM)
IF (J.NE. 2) GO TO 152
NOFET2 = NUMBER(CARD,COL,FETVC2,NOFET2)

```

SE T000010
SE T000020
SE T000030
SE T000040
SE T000050
SE T000060
SE T000070
SE T000080
SE T000090
SE T000100
SE T000110
SE T000120
SE T000130
SE T000140
SE T000150
SE T000160
SE T000170
SE T000180
SE T000190
SE T000200
SE T000210
SE T000220
SE T000230
SE T000240
SE T000250
SE T000260
SE T000270
SE T000280
SE T000290
SE T000300
SE T000310
SE T000320
SE T000330
SE T000340
SE T000350
SE T000360
SE T000370
SE T000380
SE T000390
SE T000400
SE T000410
SE T000420
SE T000430
SE T000440
SE T000450
SE T000460
SE T000470
SE T000480
SE T000490
SE T000500
SE T000510
SE T000520
SE T000530
SE T000540
SE T000550
SE T000560
SE T000570
SE T000580
SE T000590
SE T000600
SE T000610
SE T000620
SE T000630
SE T000640
SE T000650
SE T000660
SE T000670
SE T000680
SE T000690
SE T000700
SE T000710
SE T000720
SE T000730
SE T000740
SE T000750
SE T000760
SE T000770
SE T000780
SE T000790

FILE: SET13

```
CALL ORDER(FETVC2,NOFET2)
GO TO 105

C
C DATA FILE CARD
180 M = NXTCHR(CARD,COL)
  IF (M.EQ. BLNK) GO TO 105
  IF (M.EQ.U) GO TO 190
  IF (M.EQ.FF) GO TO 200
185 WRITE(6,187)
187 FORMAT(' ERROR ON DATA FILE CARD')
  GO TO 105
190 J = FIND12(CARD,COL,EQUOM)
  IF (J.NE. 2) GO TO 185
  M = NUMBER(CARD,COL,DATAPE,ZERO)
  COL = COL
  GO TO 180
200 J = FIND12(CARD,COL,EQUOM)
  IF (J.NE. 2) GO TO 185
  M = NUMBER(CARD,COL,DATFIL,ZERO)
  DATFIL = DATFIL - 1
  COL = COL - 1
  GO TO 180

C
C DOT FILE CARD
210 M = NXTCHR(CARD,COL)
  IF (M.EQ.00) GO TO 213
  IF (M.EQ.BLNK) GO TO 105
  GO TO 215
213 J = FIND12(CARD,COL,SLASH)
  IF (J.EQ. -1) GO TO 215
214 M = NXTCHR(CARD,COL)
  IF (M.EQ. BLNK) GO TO 105
  IF (M.EQ.U) GO TO 230
  IF (M.EQ.FF) GO TO 240
215 WRITE(6,220)
220 FORMAT(' ERROR ON DOT FILE CARD')
  GO TO 105
230 J = FIND12(CARD,COL,EQUOM)
  IF (J.NE. 2) GO TO 215
  M = NUMBER(CARD,COL,DOTUNT,ZERO)
  COL = COL - 1
  GO TO 214
240 J = FIND12(CARD,COL,EQUOM)
  IF (J.NE. 2) GO TO 215
  M = NUMBER(CARD,COL,DOTFIL,ZERO)
  DOTFIL = DOTFIL - 1
  COL = COL - 1
  GO TO 214

C
C OPTION CARD
330 M = NXTCHR(CARD,COL)
  IF (M.EQ. BLNK) GO TO 105
  IF (M.EQ.P) GO TO 340
  IF (M.EQ.L) GO TO 345

C*** CODE ADDED NOV 21, 1978 IN SUPPORT OF LIST PROCESSING
  IF (M.EQ.U) GO TO 350

C
333 WRITE(6,335)
335 FORMAT(' ERROR ON OPTION CARD')
  GO TO 105
340 PRKEY = 1
  GO TO 105
345 LACIE = 1
  GO TO 105
350 M = NUMBER(CARD,COL,LACIE,ZERO)
  GO TO 105

C
C DATE CARD
370 M = NXTCHR(CARD,COL)
  IF (M.EQ. BLNK) GO TO 105
  READ(30,380)DATE
  FORMAT(10X,15A4)
380 REWIND BRUNIT
```

SET00800
SET00810
SET00820
SET00830
SET00840
SET00850
SET00860
SET00870
SET00880
SET00890
SET00900
SET00910
SET00920
SET00930
SET00940
SET00950
SET00960
SET00970
SET00980
SET00990
SET01000
SET01010
SET01020
SET01030
SET01040
SET01050
SET01060
SET01070
SET01080
SET01090
SET01100
SET01110
SET01120
SET01130
SET01140
SET01150
SET01160
SET01170
SET01180
SET01190
SET01200
SET01210
SET01220
SET01230
SET01240
SET01250
SET01260
SET01270
SET01280
SET01290
SET01300
SET01310
SET01320
SET01330
SET01340
SET01350
SET01360
SET01370
SET01380
SET01390
SET01400
SET01410
SET01420
SET01430
SET01440
SET01450
SET01460
SET01470
SET01480
SET01490
SET01500
SET01510
SET01520
SET01530
SET01540
SET01550
SET01560
SET01570
SET01580

ORIGINAL PAGE IS
OF POOR QUALITY

FILE: SET13

```
C      GO TO 105
C      COMMENT CARD
C      390 M = NXTCHR(CARD,COL)
C      IF (M.EQ. BLNK) GO TO 105
C      READ(30,380)COMENT
C      REWIND RRUNIT
C      GO TO 105
C      HED1
C      400 M = NXTCHR(CARD,COL)
C      READ(30,380) HED1
C      REWIND RRUNIT
C      GO TO 105
C      HED2
C      410 M = NXTCHR(CARD,COL)
C      READ(30,380) HED2
C      REWIND RRUNIT
C      GO TO 105
C      *END*
C      420 CONTINUE
C      IF (NOFET2.NE. 0) GO TO 440
C      DO 430 I=1,30
C      FETVC2(I) = I
C      430 CONTINUE
C      NOFET2 = I
C      440 SIZE = 4 + NOFET2
C      WRITE(6,1000)
C      IF (NOFET2.NE. 0) WRITE(6,1010) (FETVC2(I),I=1,NOFET2)
C      IF (PRTKEY.EQ. 1) WRITE(6,1030)
C      1040 FORMAT(' LACIE FORMATTED DOT CARDS USED AS EOD-LARSYS FIELD CARDS'
C      )
C      IF(LACIE.EQ.1)WRITE(6,1040)
C      1000 FORMAT('/// USER HAS REQUESTED THE FOLLOWING OPTIONS :')
C      1010 FORMAT(' SELECTED CHANNELS ARE',30I3)
C      1020 FORMAT(' PRINT DATA VECTORS')
C      RETURN
C      END
```

SET01590
SET01600
SET01610
SET01620
SET01630
SET01640
SET01650
SET01660
SET01670
SET01680
SET01690
SET01700
SET01710
SET01720
SET01730
SET01740
SET01750
SET01760
SET01770
SET01780
SET01790
SET01800
SET01810
SET01820
SET01830
SET01840
SET01850
SET01860
SET01870
SET01880
SET01890
SET01900
SET01910
SET01920
SET01930
SET01940
SET01950
SET01960
SET01970
SET01980
SET01990
SET02000
SET02010
SET02020
SET02030
SET02040
SET02050
SET02060
SET02070

18. LABEL PROCESSOR

FILE LABEL

```

C      SURROUTINE LABEL (ARRAY, TOP)
      IMPLICIT INTEGER (A-Z)
      DIMENSION FLDSAV(2000), ARRAY(1)
      GO READ CONTROL CARDS
      CALL SET14 (ARRAY, TOP, EXIT)
      READ IN REQUIRED FILES
      CALL FILERO (ARRAY, TOP, NOFLD, TOTVRT, FLDSAV(1), FLDSAV(1001))
      READY TO PERFORM USER REQUESTS
      * CALL LABLR (ARRAY, TOP, NOFLD, TOTVRT, FLDSAV(1), FLDSAV(1001),
        FLDSAV(1), EXIT)
      FINISHED -- SEND* CARD
      READ(21,100) CARD
100  FORMAT(A4)
      RETURN
      END

```

```

LAB00010
LAB00020
LAB00030
LAB00040
LAB00050
LAB00060
LAB00070
LAB00080
LAB00090
LAB00100
LAB00110
LAB00120
LAB00130
LAB00140
LAB00150
LAB00160
LAB00170
LAB00180
LAB00190
LAB00200
LAB00210
LAB00220
LAB00230
LAB00240

```

ORIGINAL PAGE IS
OF POOR QUALITY

FILE ALLKIN

```

SUBROUTINE ALLKIN(DOTS,SUBVEC,SUBNO,CATVEC,MEANS,DOTSUM)
LABELS BY ALL-OF-A-KIND PROCEDURE
IMPLICIT INTEGER (A-Z)

INCLUDE COMBK1.LIST
INCLUDE COMBK4.LIST
INCLUDE COMBK15.LIST
INCLUDE COMBK6.LIST
COMMON/INFORM/NOCLS2,NOSUB2,NOFET2,VARSZ2,TOTVT2,NOFLD2,
* AVAR2,COVAR2,CLS1D2,SUBNO2,SURDS2,FLDSV2,VERTX2,
* FETVC2(30),SURVC2(75),SUBPTR(75),CLSV2(60),
* KEPPTS(60),NOGRP,GRPNAM(60),GRPDEX(61),
* GRPCHK(61),GROUPS(124)
*
* DIMENSION HED1(15),HED2(15),DATE(3),COMENT(15)
EQUIVALENCE (HED1(1),HEAD(4)),(DATE(1),HEAD(22)),
* (HED2(1),HEAD(30)),(COMENT(1),HEAD(48))
2 COMMON /LABS/NOCAT,CATNAM(60),NOCL2,CLSNM2(60),NOCAT2,CATNM2(60),
* SUBRAY(120),PTR(60),CATPTR(250),CATDOT(500),
* DOTVEC(250),COND,MIX,PROC,MAPKEY,DOTKEY,STATKY,
* SUNANG,T,NEARST,DIST,NOFEAT,FETVEC(30),MAPUN,MAPFI,
* OSAVTP,OSTAFI,NOSUN,ANGLE(8),SIZE,TOTDT2,FLDINF(6),
* CLSSYM(62),STADRS,MEANAD,TABADR,MAPADR,SUNCOR(30),
* ODOTUN,ODOTFI,MANSTA,MANDOT,DSPUNT,DSPFIL,DSPKEY,PRNSTS,
* PRNDOT,FLDNAM,VERTEX(22),NOVRT,NSUN,ANGLES(8)
* TOTDT3,FLDADR,VIXADR
COMMON/GLOBAL/HEAD(63),MAPTAP,DATAP,SAVTAP,BMFILE,BMKEY,
* HISFIL,HISKEY,TRFORM,ERIPTP,ERPKEY,MAPUNT,NOFILE,
* DRUMAD,DRMWDS,PAGSIZ,DATFIL,STAFIL,ASAV,ASAVFL
* NHSTUN,NHSTFI,SCTRUN,MAPFIL
* DOTUNT,DOTFIL,NCHPAS,TRNSFL,BMTRFL,HISTFL,PCHUNT,
* CRDUNT,PRUNT,RANDIO
CSEND
* DIMENSION DOTS(SIZE,TOTDT2),SUBVEC(1),SUBNO(1),CATVEC(1),
* CATGRY(60),CLSTNO(250),DOTNUM(250),
* DISTNC(250)
* DIMENSION MEANS(NOFET2,1)
* DIMENSION TIES(250),DOTSUM(60,60)
RETRIEVE CLUSTER CLASSIFICATION FOR EACH DOT
NSAMP = (FLDINF(5)-FLDINF(4))/FLDINF(6) + 1
LINES = (FLDINF(2) - FLDINF(2))/FLDINF(3) + 1
DO 100 I=1,TOTDT2
  ILINE = DOTS(2,I)
  ISAMP = DOTS(1,I)
  PIXADR = DRUMAD + (ILINE-1)*NSAMP + ISAMP-1
  CALL RREAD(PIXADR,NUMBER,1,ISTAT)
C
CLSTNO -- CONTAINS THE CLUSTER NO. OF EACH DOT
90 IF (ISTAT.EQ. 1) GO TO 90
100 CLSTNO(I) = NUMBER
C
  WRITE(6,HEAD)
  WRITE(6,111)
111  FORMAT(150,'LABELING BY ALL-OF-A-KIND PROCEDURE')
  WRITE(6,1111)
1111 FORMAT(20X,'CLUSTER LABELING DETAILS',/)
  WRITE(6,1000)
1000 FORMAT(3X,'CLUSTER',2X,'CLUSTER',3X,'DOT',4X,'DOT',
1 7X,'DOT',6X,'DOT',7,3X,'NUMBER',4X,'LABEL',3X,'LABEL',
2 2X,'NUMBER',2X,'DISTANCE',2X,'CLUSTER',/)
C
  MAJOR DO LOOP
C
DO 300 I=1,NOSUB2
  REINITIALIZE
C
DO 155 J=1,NOCAT
155 CATGRY(J) = 0
  MAX = 0
  NODOT = 0
  TIE = 0

```

FILE ALLKIN

```

C      FIND ALL DOTS IN CLUSTER I
C      DO 150 J=1,TOTDT2
C      IF (CLSTNO(J).NE. I) GO TO 150
C      K = DOTS(4,J)
C      CATGRY(K) = CATGRY(K) + 1
C      NODOT = NODOT + 1
C      DOTNUM(NODOT) = J
150    CONTINUE
C      WERE DOTS - 1 OF A KIND
C      WERE DOTS- MEXED
C      WERE DOTS-- NULL SET
C      DO 160 J=1,NOCAT
C      IF (CATGRY(J).LT. MAX) GO TO 160
C      MAX = CATGRY(J)
C      CATNUM = J
160    CONTINUE
C      IF (MAX .NE. 0) GO TO 170
C      DEFAULT TO K-NEAREST NEIGHBOR
C      ITER = 1
C      TAB1 = TABADR + (I-1)*TOTDT2
C      SWITCH = 1
C      WRITE(6,165) I
165    FORMAT(/,3X,1** DEFAULTING TO K-NEAREST NEIGHBOR PROCEDURE FOR CL
C      USTER **,15)
C      CALL KNEAR(DOTS,DUM,DUM,CATVEC,ITER,TAB1,SWITCH,CATNUM,I,MEANS,
C      1 DOTSUM)
C      GO TO 300
C      LABEL BY MAJORITY
C      170 CONTINUE
C      PRINT CLUSTER INFORMATION
C      TAB1 = TABADR + (I-1)*TOTDT2
C      CALL RREAD(TAB1,DISTNC,TOTDT2,1STAT)
C      GO TO CODE FOR A TIE
C      GO TO 390
C      WRITE DOT DETAILS
C      175 WRITE(6,1100) I,CATNAM(CATNUM)
1100    FORMAT(/,5X,12,8X,1A4)
C      DO 110 J=1,NODOT
C      DST=DOTNUM(J)
C      K=DOTS(4,DST)
C      IF (J.EQ.1) WRITE(6,1201) CATNAM(K),DOTVEC(DST),DISTNC(DST).
1201    1 CLSTNO(DST)
C      FORMAT(1H,12,1A4,2X,13,4X,F7.2,4X,12)
C      IF (J.GT.1) WRITE(6,1200) CATNAM(K),DOTVEC(DST),DISTNC(DST).
1200    1 CLSTNO(DST)
C      FORMAT(23X,1A4,2X,13,4X,F7.2,4X,12)
C      DOTSUM(I,K)=DOTSUM(I,K)+1
110    CONTINUE
C      IF (TIE .EQ. 0) GO TO 300
C      WRITE(6,211)
211    FORMAT(23X,1A TIE OCCURRED. THE FOLLOWING DOT(S) WERE DISCARDED
C      1 //)
C      DO 213 J=1,TIE
C      INDEX1 = TIES(J)
C      LABELS = DOTS(4,INDEX1)
213    WRITE(6,1200) CATNAM(LABELS),DOTVEC(INDEX1),DISTNC(INDEX1),

```

ALL00770
 ALL00780
 ALL00790
 ALL00800
 ALL00810
 ALL00820
 ALL00830
 ALL00840
 ALL00850
 ALL00860
 ALL00870
 ALL00880
 ALL00890
 ALL00900
 ALL00910
 ALL00920
 ALL00930
 ALL00940
 ALL00950
 ALL00960
 ALL00970
 ALL00980
 ALL00990
 ALL01000
 ALL01010
 ALL01020
 ALL01030
 ALL01040
 ALL01050
 ALL01060
 ALL01070
 ALL01080
 ALL01090
 ALL01100
 ALL01110
 ALL01120
 ALL01130
 ALL01140
 ALL01150
 ALL01160
 ALL01170
 ALL01180
 ALL01190
 ALL01200
 ALL01210
 ALL01220
 ALL01230
 ALL01240
 ALL01250
 ALL01260
 ALL01270
 ALL01280
 ALL01290
 ALL01300
 ALL01310
 ALL01320
 ALL01330
 ALL01340
 ALL01350
 ALL01360
 ALL01370
 ALL01380
 ALL01390
 ALL01400
 ALL01410
 ALL01420
 ALL01430
 ALL01440
 ALL01450
 ALL01460
 ALL01470
 ALL01480
 ALL01490
 ALL01500
 ALL01510
 ALL01520

FILE ALLKIN

| | | |
|------|---|----------|
| C | 1 CLSTNO(INDEX1) | ALL01530 |
| C | 300 CONTINUE | ALL01540 |
| C | WRITE DOT SUMMARY | ALL01550 |
| C | WRITE(6,2222) | ALL01560 |
| 2222 | FORMAT(1H1,20X,'CLUSTER LABELING SUMMARY',/) | ALL01570 |
| | WRITE(6,1300) | ALL01580 |
| 1300 | FORMAT(3X,'CLUSTER',20X,'NUMBER OF DOTS USED (BY CATEGORY NAME)') | ALL01590 |
| | ISTR=1 | ALL01600 |
| | IEND=NOCAT | ALL01610 |
| | IF(IEND.GT.15) IEND=15 | ALL01620 |
| | WRITE(6,1305) | ALL01630 |
| 1305 | FORMAT(3X,'NUMBER',3X,'LABEL',7X,50(1H-)) | ALL01640 |
| | WRITE(6,1330) | ALL01650 |
| 1330 | FORMAT(1H+,T20,'TOTAL') | ALL01660 |
| | WRITE(6,1310) (CATNAM(IJ),IJ=1,IEND) | ALL01670 |
| 1310 | FORMAT(30X,1A4,14(3X,1A4)) | ALL01680 |
| | DO 600 I=1,NOSUB2 | ALL01690 |
| | TOTAL=0 | ALL01700 |
| | DO 550 J=1,NOCAT | ALL01710 |
| 550 | TOTAL=TOTAL+DOTSUM(I,J) | ALL01720 |
| | K=CATVEC(I) | ALL01730 |
| | WRITE(6,1320) I,CATNAM(K),TOTAL,(DOTSUM(I,J),J=1,IEND) | ALL01740 |
| 1320 | FORMAT(//,5X,12,6X,1A4,15,2X,15(2X,15)) | ALL01750 |
| 600 | CONTINUE | ALL01760 |
| 602 | IF(IEND.EQ.NOCAT) GO TO 650 | ALL01770 |
| | ISTR=IEND+1 | ALL01780 |
| | IEND=NOCAT | ALL01790 |
| | IF(IEND.GT.ISTR+14) IEND=ISTR+14 | ALL01800 |
| | WRITE(6,1350) | ALL01810 |
| 1350 | FORMAT(//) | ALL01820 |
| | WRITE(6,1300) | ALL01830 |
| | WRITE(6,1305) | ALL01840 |
| | WRITE(6,1310) (CATNAM(IJ),IJ=ISTR,IEND) | ALL01850 |
| | DO 610 I=1,ITER | ALL01860 |
| | K=CATVEC(I) | ALL01870 |
| | WRITE(6,1340) I,CATNAM(K),(DOTSUM(I,J),J=ISTR,IEND) | ALL01880 |
| 1340 | FORMAT(//,5X,12,6X,1A4,7X,15(2X,15)) | ALL01890 |
| 610 | CONTINUE | ALL01900 |
| | GO TO 602 | ALL01910 |
| C | GROUP LABELED CLUSTERS | ALL01920 |
| C | K = 0 | ALL01930 |
| 650 | DO 310 I=1,NOCAT | ALL01940 |
| | DO 310 J=1,NOSUB2 | ALL01950 |
| | IF (CATVEC(J) .NE. I) GO TO 310 | ALL01960 |
| | SUBNO(I) = SUBNO(I) + 1 | ALL01970 |
| | K = K + 1 | ALL01980 |
| | SUBVEC(K) = J | ALL01990 |
| 310 | CONTINUE | ALL02000 |
| | RETURN | ALL02010 |
| C | CODE FOR A TIE | ALL02020 |
| C | 390 IF (MAX .EQ. 0) GO TO 175 | ALL02030 |
| | DO 400 II=1,NOCAT | ALL02040 |
| | IF (II .EQ. CATNUM) GO TO 400 | ALL02050 |
| | IF (MAX .EQ. CATGRY(II)) GO TO 410 | ALL02060 |
| 400 | CONTINUE | ALL02070 |
| | CATVEC(I) = CATNUM | ALL02080 |
| | GO TO 175 | ALL02090 |
| C | 410 TIE = TIE + 1 | ALL02100 |
| | MAXDST = 0 | ALL02110 |
| | DO 420 J=1,NODOT | ALL02120 |
| | NO = DOTNU-1(J) | ALL02130 |
| | DST = DISTNC(NO) | ALL02140 |
| | IF (MAXDST .GT. DST) GO TO 420 | ALL02150 |
| | DOTNO = NO | ALL02160 |
| | MAXDST = DST | ALL02170 |
| | INDEX = J | ALL02180 |
| | LABEL = DOTS(4,NO) | ALL02190 |
| 420 | CONTINUE | ALL02200 |
| | | ALL02210 |
| | | ALL02220 |
| | | ALL02230 |
| | | ALL02240 |
| | | ALL02250 |
| | | ALL02260 |
| | | ALL02270 |
| | | ALL02280 |

FILE ALLKIN

```
C      TIES(TIE) = DOTNO
C      CATGRY(LABEL) = CATGRY(LABEL) - 1
      NODOT = NODOT - 1
      IF (INDEX.EQ. (NODOT + 1)) GO TO 445
C      DO 440 II=INDEX,NODOT
C 440 DOTNUM(II) = DOTNUM(II+1)
C 445 CONTINUE
      MAX = 0
      DO 470 II=1,NOCAT
      IF (CATGRY(II) .LE. MAX) GO TO 470
      MAX = CATGRY(II)
      CATNUM = II
C 470 CONTINUE
      GO TO 390
      END
```

```
ALL02290
ALL02300
ALL02310
ALL02320
ALL02330
ALL02340
ALL02350
ALL02360
ALL02370
ALL02380
ALL02390
ALL02400
ALL02410
ALL02420
ALL02430
ALL02440
ALL02450
ALL02460
ALL02470
```


FILE: ASCEND

| | | |
|----|--|----------|
| | SUBROUTINE ASCEND(SCN,LNCAT,PTR1,PTR2) | ASC00010 |
| | IMPLICIT INTEGER(A-X) | ASC00020 |
| | DIMENSION PTR1(LNCAT),PTR2(LNCAT) | ASC00030 |
| | REAL SCN(LNCAT),SAVE | ASC00040 |
| | J=0 | ASC00050 |
| 60 | J=J+1 | ASC00060 |
| | IF(J.GT.LNCAT)GO TO 90 | ASC00070 |
| | IF(J.EQ.LNCAT)GO TO 75 | ASC00080 |
| | IF (SCN(J) .GT. SCN(J+1)) GO TO 70 | ASC00090 |
| | GO TO 60 | ASC00100 |
| C | | ASC00110 |
| 70 | SAVE=SCN(J) | ASC00120 |
| | SCN(J)=SCN(J+1) | ASC00130 |
| | SCN(J+1)=SAVE | ASC00140 |
| C | | ASC00150 |
| | SAVE1=PTR1(J) | ASC00160 |
| | PTR1(J)=PTR1(J+1) | ASC00170 |
| | PTR1(J+1)=SAVE1 | ASC00180 |
| C | | ASC00190 |
| | SAVE2=PTR2(J) | ASC00200 |
| | PTR2(J)=PTR2(J+1) | ASC00210 |
| | PTR2(J+1)=SAVE2 | ASC00220 |
| 75 | K=J | ASC00230 |
| A0 | IF (K.EQ.1)GO TO 40 | ASC00240 |
| | IF (SCN(K) .GT. SCN(K-1)) GO TO 60 | ASC00250 |
| C | | ASC00260 |
| | SAVE=SCN(K-1) | ASC00270 |
| | SCN(K-1)=SCN(K) | ASC00280 |
| | SCN(K)=SAVE | ASC00290 |
| C | | ASC00300 |
| | SAVE1=PTR1(K-1) | ASC00310 |
| | PTR1(K-1)=PTR1(K) | ASC00320 |
| | PTR1(K)=SAVE1 | ASC00330 |
| C | | ASC00340 |
| | SAVE2=PTR2(K-1) | ASC00350 |
| | PTR2(K-1)=PTR2(K) | ASC00360 |
| | PTR2(K)=SAVE2 | ASC00370 |
| | K=K-1 | ASC00380 |
| | GO TO A0 | ASC00390 |
| 90 | CONTINUE | ASC00400 |
| | RETURN | ASC00410 |
| | END | ASC00420 |

FILE: CLRKEY

| | |
|---|--|
| <pre> C SUBROUTINE CLRKEY(XSIZ, IDATA, NOSUB2, CH, MEANS, NC) C* C* CLKYS ADDS THE COLOR KEYS TO A UNIVERSAL FORMAT TAPE C* THE COLORS ARE OUTPUT AS SQUARES IMAGES (10X10) C* IMPLICIT INTEGER (A-Z) C* DIMENSION IDATA(XSIZ, CH) C* REAL MEANS(NC, NOSUB2) C C LSTLIN = 0 C LINE = 0 C TOTKEY = 0 C NKFYS = XSIZ/11 C NOKEY = NOSUB2 C C 90 DO 100 J=1, CH C* WRITE A SCAN LINE OF ZEROS - USED FOR SEPARATING THE THE COLORS C* C DO 100 I=1, XSIZ C 100 IDATA(I, J) = 0 C C CALL WRTLN(IDATA, LSTLIN) C LINE = LINE + 1 C C 110 IF (NKEYS .LE. NOKEY) NOKEY = NKEYS C KK = 0 C C DO 150 I=1, NOKEY C TOTKEY = TOTKEY + 1 C DO 140 J=1, NC C DO 130 K=1, 10 C KK = (I-1)*11 + K C 130 IDATA(KK, J) = MEANS(J, TOTKEY) + 0.5 C 140 CONTINUE C* C* WRITE A SCAN LINE OF COLORS C* C 150 CONTINUE C NOKEY = NOSUB2 - TOTKEY C C DO 160 I=1, 10 C IF (NOKEY .LE. 0 .AND. I .EQ. 10) LSTLIN = -1 C 160 CALL WRTLN(IDATA, LSTLIN) C LINE = LINE + 10 C IF (NOKEY .LE. 0) GO TO 170 C GO TO 90 C C 170 CONTINUE C C RETURN C END </pre> | <pre> CLR00010 CLR00020 CLR00030 CLR00040 CLR00050 CLR00060 CLR00070 CLR00080 CLR00090 CLR00100 CLR00110 CLR00120 CLR00130 CLR00140 CLR00150 CLR00160 CLR00170 CLR00180 CLR00190 CLR00200 CLR00210 CLR00220 CLR00230 CLR00240 CLR00250 CLR00260 CLR00270 CLR00280 CLR00290 CLR00300 CLR00310 CLR00320 CLR00330 CLR00340 CLR00350 CLR00360 CLR00370 CLR00380 CLR00390 CLR00400 CLR00410 CLR00420 CLR00430 CLR00440 CLR00450 CLR00460 CLR00470 CLR00480 CLR00490 CLR00500 CLR00510 CLR00520 </pre> |
|---|--|

ORIGINAL PAGE IS
OF POOR QUALITY

FILE: CLSMAP

```

SURROUTINE CLSMAP(CATSUB,SWTCH,SUBNO,SUBVEC,SUBDES,CATVEC)
OUTPUTS LINE PRINTER MAP(CONDITIONAL AND MIXED)
OUTPUTS MAPFIL TAPE
SWTCH = 1 -- COND. MAP
SWTCH = 2 -- MIXED MAP
IMPLICIT INTEGER (A-Z)
INCLUDE COMRK1.LIST
INCLUDE COMRK15.LIST
INCLUDE COMRK6.LIST
INCLUDE COMRK4.LIST
COMMON/INFORM/NOCL2,NOSUR2,NOFET2,VARSZ2,TOTVT2,NOFLD2,
* AVAR2,COVAR2,CLSID2,SUBNO2,SURDS2,FLOSV2,VERTX2,
* FETVC2(30),SURVC2(75),SUBPTR(75),CLSV2(60),
* KFPPTS(60),NOGRP,GHPNAM(60),GRPDEX(61),
* GRPCHK(61),GROUPS(124)
COMMON /LABS/NOCAT,CATNAM(60),NOCL2,CLSNM2(60),NOCAT2,CATNM2(60),
* SURRAY(120),PTR(60),CATPTR(250),CATDOT(500),
* DOTVEC(250),COND,MIX,PROC,MAPKEY,DOTKEY,STATKY,
* SUNANG,T,NEARST,DIST,NOFEAT,FETVEC(30),OMAPUN,OMAPFI,
* OSAVTP,OSTAFI,NOSUN,ANGLE(8),SIZE,TOTDT2,FLOINF(4),
* CLSSYM(62),STADRS,MEANAD,TABADR,MAPADR,SUNCOR(30),
* ODOTIUM,ODOTFI,MANSTA,MANDOT,DSPUNT,DSPFIL,DSPKEY,PRNSTS,
* PRNDOT,FLOINAM,VERTEX(22),NOVRT,NSUN,ANGLES(8)
* TOTDT3,FLOADR,VTXADR
COMMON/GLOBAL/HEAD(63),MAPTAP,DATAP,SAVTAP,BMFILE,BMKEY,
* HISFIL,HISKEY,TRFORM,ERPTP,ERPKEY,MAPUNT,NOFILE,
* DRUMAD,DRMADS,PAGSIZ,DATFIL,STAFIL,ASAV,ASAVFL
* NHSTUN,NHSTFI,STRUN,MAPFIL
* DOTUNT,DOTFIL,NCHPAS,TRNSFL,BMTRFL,HISTFL,PCHUNT,
* CPDUNT,PHTUNT,MAND10
DIMENSION HED1(15),HED2(15),DATE(3),COMENT(15)
EQUIVALENCE (HED1(1),HEAD(4)),(DATE(1),HEAD(22)),
2 (HED2(1),HEAD(30)),(COMENT(1),HEAD(48))
CSEND
DIMENSION SURVEC(1)
DIMENSION TITLE(3,2), COL(3,110)
DIMENSION SYMBOL(62)
DATA TITLE/COND,ITIO,INAL,MI,MIXED,/,
DATA SYMBOL/1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49,50,51,52,53,54,55,56,57,58,59,60,61,62,63,64,65,66,67,68,69,70,71,72,73,74,75,76,77,78,79,80,81,82,83,84,85,86,87,88,89,90,91,92,93,94,95,96,97,98,99,100,101,102,103,104,105,106,107,108,109,110,111,112,113,114,115,116,117,118,119,120,121,122,123,124,125,126,127,128,129,130,131,132,133,134,135,136,137,138,139,140,141,142,143,144,145,146,147,148,149,150,151,152,153,154,155,156,157,158,159,160,161,162,163,164,165,166,167,168,169,170,171,172,173,174,175,176,177,178,179,180,181,182,183,184,185,186,187,188,189,190,191,192,193,194,195,196,197,198,199,200,201,202,203,204,205,206,207,208,209,210,211,212,213,214,215,216,217,218,219,220,221,222,223,224,225,226,227,228,229,230,231,232,233,234,235,236,237,238,239,240,241,242,243,244,245,246,247,248,249,250,251,252,253,254,255,256,257,258,259,260,261,262,263,264,265,266,267,268,269,270,271,272,273,274,275,276,277,278,279,280,281,282,283,284,285,286,287,288,289,290,291,292,293,294,295,296,297,298,299,300,301,302,303,304,305,306,307,308,309,310,311,312,313,314,315,316,317,318,319,320,321,322,323,324,325,326,327,328,329,330,331,332,333,334,335,336,337,338,339,340,341,342,343,344,345,346,347,348,349,350,351,352,353,354,355,356,357,358,359,360,361,362,363,364,365,366,367,368,369,370,371,372,373,374,375,376,377,378,379,380,381,382,383,384,385,386,387,388,389,390,391,392,393,394,395,396,397,398,399,400,401,402,403,404,405,406,407,408,409,410,411,412,413,414,415,416,417,418,419,420,421,422,423,424,425,426,427,428,429,430,431,432,433,434,435,436,437,438,439,440,441,442,443,444,445,446,447,448,449,450,451,452,453,454,455,456,457,458,459,460,461,462,463,464,465,466,467,468,469,470,471,472,473,474,475,476,477,478,479,480,481,482,483,484,485,486,487,488,489,490,491,492,493,494,495,496,497,498,499,500,501,502,503,504,505,506,507,508,509,510,511,512,513,514,515,516,517,518,519,520,521,522,523,524,525,526,527,528,529,530,531,532,533,534,535,536,537,538,539,540,541,542,543,544,545,546,547,548,549,550,551,552,553,554,555,556,557,558,559,560,561,562,563,564,565,566,567,568,569,570,571,572,573,574,575,576,577,578,579,580,581,582,583,584,585,586,587,588,589,590,591,592,593,594,595,596,597,598,599,600,601,602,603,604,605,606,607,608,609,610,611,612,613,614,615,616,617,618,619,620,621,622,623,624,625,626,627,628,629,630,631,632,633,634,635,636,637,638,639,640,641,642,643,644,645,646,647,648,649,650,651,652,653,654,655,656,657,658,659,660,661,662,663,664,665,666,667,668,669,670,671,672,673,674,675,676,677,678,679,680,681,682,683,684,685,686,687,688,689,690,691,692,693,694,695,696,697,698,699,700,701,702,703,704,705,706,707,708,709,710,711,712,713,714,715,716,717,718,719,720,721,722,723,724,725,726,727,728,729,730,731,732,733,734,735,736,737,738,739,740,741,742,743,744,745,746,747,748,749,750,751,752,753,754,755,756,757,758,759,760,761,762,763,764,765,766,767,768,769,770,771,772,773,774,775,776,777,778,779,780,781,782,783,784,785,786,787,788,789,790,791,792,793,794,795,796,797,798,799,800,801,802,803,804,805,806,807,808,809,810,811,812,813,814,815,816,817,818,819,820,821,822,823,824,825,826,827,828,829,830,831,832,833,834,835,836,837,838,839,840,841,842,843,844,845,846,847,848,849,850,851,852,853,854,855,856,857,858,859,860,861,862,863,864,865,866,867,868,869,870,871,872,873,874,875,876,877,878,879,880,881,882,883,884,885,886,887,888,889,890,891,892,893,894,895,896,897,898,899,900,901,902,903,904,905,906,907,908,909,910,911,912,913,914,915,916,917,918,919,920,921,922,923,924,925,926,927,928,929,930,931,932,933,934,935,936,937,938,939,940,941,942,943,944,945,946,947,948,949,950,951,952,953,954,955,956,957,958,959,960,961,962,963,964,965,966,967,968,969,970,971,972,973,974,975,976,977,978,979,980,981,982,983,984,985,986,987,988,989,990,991,992,993,994,995,996,997,998,999,1000
C
DIMENSION CATSUB(1),SUBNO(1),IR(1000),OUT(1000),COLORS(62)
DIMENSION DATA(1000),RELCLR(60)
REAL RELCLR
DIMENSION CATVEC(1)
DIMENSION SUBDES(60)
DIMENSION COLOR(64),COLATE(60),DELETE(60)
DATA COLOR/1,207,79,111,47,175,143,71,
* 167,105,107,171,199,135,39,71,
* 205,137,75,41,45,69,73,75,
* 77,133,137,139,141,33,35,37,
* 39,41,43,45,47,103,105,107,
* 109,197,201,203,205,169,171,173,
* 131,135,137,141,67,71,75,77,
* 161,163,165,167,169,171,225,239/
DATA PRND/1,1/
COLORS(NOSUR2 + 1) = COLOR(1)
COLORS(NOSUR2 + 2) = COLOR(1)
C CHECK FOR DELETED CATEGORIES
KEPT = 0
DO 2 II = 1,NOCAT
DELETE(II) = 0
DO 1 I = 1,NOSUR2
IF (CATSUB(I).EQ.II) DELETE(II) = 1
CONTINUE
KEPT = KEPT + DELETE(II)
CONTINUE
IF (KEPT.EQ.0) GO TO 7
II = 1
SMALLS = 0

```

FILE: CLSMAP

```

3  SMALL = 2**30 + (2**30-2)
   DO 4 I = 1,NOCAT
     DUM = CATNAM(I)-2**31
     IF (DUM.LE.SMALLS) GO TO 4
     IF (DUM.GE.SMALL) GO TO 4
     III = I
     SMALL = DUM
4  CONTINUE
   COLATE(III) = II
   SMALLS = SMALL
   IF (DELETE(III).EQ.0) GO TO 6
   II = II + 1
   IF (II.LE.KEPT) GO TO 3
6  CONTINUE
7  ASSIGN CATEGORY SYMBOL TO EACH CLUSTER
   DO 10 I=1,NOSUB2
     CATNUM = CATSUB(I)
     CLSSYM(I) = SYMBOL(CATNUM)
     IF (CATNUM.GT.NOCAT) GO TO 9
     DUM = COLATE(CATNUM)
     DELETE(I) = DUM
     COLORS(I) = COLOR(DUM + 1)
     GO TO 10
9  COLORS(I) = COLOR(64 + KEPT - CATNUM)
   DELETE(I) = 63 + KEPT - CATNUM
10 CONTINUE
    CLSSYM (NOSUB2 + 1) = IPND
    CLSSYM (NOSUB2 + 2) = IPND
    CALL SFTHMG(68,0,68)
    WRITE(6,HEAD)
C  CALL MAPHND(NOCAT,CLSSYM,CATNAM,CATVEC,SUBDES,CATSUB)
C  WRITE(6,5)(TITLE(I,SWTCH),I=1,3)
C  5  FORMAT(/750X,344,'CLUSTER MAP',/)
C  ISTRT = FLDINF(4)
C  IEND = FLDINF(5)
C  SAMINC = FLDINF(6)
C  LININC = FLDINF(3)
C  LINSTR = FLDINF(1)
C  LINEND = FLDINF(2)
C  ILINE = (LINEND - LINSTR)/LININC + 1
C  NSAMP = (IEND-ISTRT)/SAMINC + 1
C  PTS = 0
C  POSITION TAPE
C  REWIND OMAPUN
C  IF (OMAPFI.NE.0) CALL FSFMLE(OMAPUN,OMAPFI,ISTAT)
C  CALL WRTHED(1,1,NSAMP,1,OMAPUN)
C  PRINT LINE PRINTER MAP
C  IPFLAG = 1
C  GO TO 400
14 CONTINUE
   II = 0
C  IPTS = NSAMP
   IF (IPTS.GT.110) IPTS = 110
   IF (NSAMP.LE.110) GO TO 15
   IPD = NSAMP - 110
15 CONTINUE
   DO 300 I=LINSTR,LINEND,LININC
     II = II + 1
     PIXADR = DRUMAD + (II-1)*NSAMP
     CALL RREAD(PIXADR,IR,NSAMP,ISTAT)
20 IF (ISTAT.EQ.1) GO TO 20
C  DRUM ADDRESS MAPADR
C  DO 30 J=1,NSAMP
   L = IP(J)
   IF (L.NE.0) GO TO 25
   INDATA(J) = 0

```

CLS00800
 CLS00810
 CLS00820
 CLS00830
 CLS00840
 CLS00850
 CLS00860
 CLS00870
 CLS00880
 CLS00890
 CLS00900
 CLS00910
 CLS00920
 CLS00930
 CLS00940
 CLS00950
 CLS00960
 CLS00970
 CLS00980
 CLS00990
 CLS01000
 CLS01010
 CLS01020
 CLS01030
 CLS01040
 CLS01050
 CLS01060
 CLS01070
 CLS01080
 CLS01090
 CLS01100
 CLS01110
 CLS01120
 CLS01130
 CLS01140
 CLS01150
 CLS01160
 CLS01170
 CLS01180
 CLS01190
 CLS01200
 CLS01210
 CLS01220
 CLS01230
 CLS01240
 CLS01250
 CLS01260
 CLS01270
 CLS01280
 CLS01290
 CLS01300
 CLS01310
 CLS01320
 CLS01330
 CLS01340
 CLS01350
 CLS01360
 CLS01370
 CLS01380
 CLS01390
 CLS01400
 CLS01410
 CLS01420
 CLS01430
 CLS01440
 CLS01450
 CLS01460
 CLS01470
 CLS01480
 CLS01490
 CLS01500
 CLS01510
 CLS01520
 CLS01530
 CLS01540
 CLS01550
 CLS01560
 CLS01570
 CLS01580

ORIGINAL PAGE IS
 OF POOR QUALITY

FILE: CLSMAP

```

GO TO 30
25 OUT(J) = CLSSYM(L)
30 IDATA(J) = COLORS(L)

WRITE IDATA OUT TO TAPE
CALL WRTLN(IDATA,LSTLIN)

WRITE REMAINDER OF PIXELS ON DRUM FOR SUBSEQUENCE WRITING

MAPDRM = MAPADR + (II-1)*IPD
CALL RWRITE(MAPDRM,OUT(11),IPD,ISTAT)
50 WRITE(6,60) I, (OUT(IK), IK=1, IPTS)
60 FORMAT(2X, I5, 2X, 110A1)
300 CONTINUE
XSIZ = NSAMP
CH = 1
DO #0 I = 1, NOSUB2
RELCLR(I) = COLORS(I)
40 CONTINUE
CALL CLRKEY(XSIZ, IDATA, NOSUB2, CH, RELCLR, CH)
ISTART = 1
IPD = IPD
305 I = 0
PTS = PTS + IPTS
IF (PTS .GE. NSAMP) GO TO 360
IF (IPD .GT. 110) IPTS = 110
IF (IPD .LE. 110) IPTS = IPD
IENDS = ISTART + IPTS - 1
IPD = IPD - IPTS

PRINT REST OF MAP

IPFLAG = 2
GO TO 500
308 CONTINUE

DO 350 I=LINSTR, LINEND, LININC
II = II + 1
MAPDRM = MAPADR + (II-1)*IPD
CALL RREAD(MAPDRM, OUT, IPD, ISTAT)
310 IF (ISTAT .EQ. 1) GO TO 310
350 WRITE(6,240) I, (OUT(IK), IK=ISTART, IENDS)
240 FORMAT(2X, I6, 1X, 110A1)
ISTART = IENDS + 1
GO TO 305

FINISHED

360 CONTINUE
WRITE(6,370)
370 FORMAT(1H, 'NEW ORDERING AND COLOR KEY CODES')
WRITE(6,375)
375 FORMAT('/// 23X, ' OLD NEW CAT COLOR ORDER COLORS'///)
DO 11 I = 1, NOSUB2
WRITE(6,340) I, CATSUB(I), DELETE(I), COLORS(I)
340 FORMAT(20X, 4(16, 6X))
CONTINUE
11 CALL SETMRG(68, 4, 62)
RETURN

500 CONTINUE
J = 0
DO #1 IJ = ISTART, IEND, SAMINC
J = J + 1
COL(1, J) = IJ/100
COL(2, J) = MOD(IJ, 100)/10
COL(3, J) = MOD(IJ, 10)
IF (IJ .EQ. 110) GO TO #2
#1 CONTINUE
#2 SAMEN = IJ
ISTRT = SAMEN + SAMINC
JPTS = J
WRITE(6, 95)
DO #3 IJ = 1, 3
#3 WRITE(6, 90) (COL(IJ, J), J = 1, JPTS)
90 FORMAT(9X, 110I1)
WRITE(6, 95)
95 FORMAT(/)

```

CL501590
CL501600
CL501610
CL501620
CL501630
CL501640
CL501650
CL501660
CL501670
CL501680
CL501690
CL501700
CL501710
CL501720
CL501730
CL501740
CL501750
CL501760
CL501770
CL501780
CL501790
CL501800
CL501810
CL501820
CL501830
CL501840
CL501850
CL501860
CL501870
CL501880
CL501890
CL501900
CL501910
CL501920
CL501930
CL501940
CL501950
CL501960
CL501970
CL501980
CL501990
CL502000
CL502010
CL502020
CL502030
CL502040
CL502050
CL502060
CL502070
CL502080
CL502090
CL502100
CL502110
CL502120
CL502130
CL502140
CL502150
CL502160
CL502170
CL502180
CL502190
CL502200
CL502210
CL502220
CL502230
CL502240
CL502250
CL502260
CL502270
CL502280
CL502290
CL502300
CL502310
CL502320
CL502330
CL502340
CL502350
CL502360
CL502370

FILE: CLSNAP

IF (IPFLAG .EQ. 1) GO TO 14
IF (IPFLAG .EQ. 2) GO TO 308
END

CLS02380
CLS02390
CLS02400

ORIGINAL PAGE IS
OF POOR QUALITY

FILE: CNDMAP

```

C      SURROUTINE CNDMAP(DOTS,CNDSUR,CATVEC)
C      FLAGS THE CONDITIONAL CLUSTERS
C      IMPLICIT INTEGER (A-Z)
C      DIMENSION CNDSUR(60),DISTNC(250),DOTS(SIZE,TOTDT2),CATVEC(1)
C      REAL DSTN,DISTNC
C      INCLUDE COMCHK1.LIST
C      INCLUDE CMH#15.LIST
C      COMMON/INFORM/NOCLS2,NOSUR2,NOFET2,VAR522,TOTVT2,NOFLD2,
C      AVAR2,CNVAR2,CLSID2,SUBNO2,SUBDS2,FLDSV2,VERTX2,
C      FETVC2(30),SURVC2(75),SURPTR(75),CLSV2(60),
C      KEPPTS(60),NOGRP,GRPNAM(60),GRPDEX(61),
C      GRPCHK(41),GROUPS(124)
C      COMMON /LABS/NOCAT,CATNAM(60),NOCL2,CLSNM2(60),NOCAT2,CATNM2(60),
C      SURRAY(120),PTR(60),CATPTR(250),CATDOT(500),
C      DOTVEC(250),COND,MIX,PROC,MAPKEY,DOTKEY,STATKY,
C      SUNANG,T,NFARST,DIST,NOFEAT,FETVEC(30),OMAPUN,OMAPF1,
C      OSAVTP,OSTAFI,NOSUN,ANGLE(8),SIZE,TOTDT2,FLDINF(6),
C      CLSSYM(62),STADRS,MEANAD,TABADR,MAPADR,SUNCOR(30),
C      ODOTUN,ODOTF1,MANSTA,MANDOT,OSPUNT,OSPFIL,OSPKEY,PRNSTS,
C      PRNIOT,FLDNAM,VERTX(22),NOVRT,NSUN,ANGLES(8)
C      TOTDT3,FLNADR,VTXADH
CSEND
C      FIND ALL CONDITIONAL CLUSTERS
C      NEXT = 63
C      DO 100 I=1,NOSUR2
C      CNDSUR(I) = CATVEC(I)
C      DO 200 J=1,NOSUB2
C      TAB1 = TABADR + (I-1)*TOTDT2
C      CALL RREAD(TAB1,DISTNC,TOTDT2,ISTAT)
C      110 IF (ISTAT.EQ. 1) GO TO 110
C      DSTN = 255.
C      LABEL = CATVEC(I)
C      DO 120 J=1,TOTDT2
C      IF (DSTN.LT. DISTNC(J)) GO TO 120
C      IF (DOTS(4,J).NE. LABEL) GO TO 120
C      DSTN = DISTNC(J)
C      120 CONTINUE
C      COMPARE THRESHOLD VALUE T
C      IF (DSTN.LF. T) GO TO 200
C      FLAG AS CONDITIONAL
C      NEXT = NEXT + 1
C      CNDSUR(I) = NEXT
C      200 CONTINUE
C      RETURN
C      END

```

CND00010
CND00020
CND00030
CND00040
CND00050
CND00060
CND00070
CND00080
CND00090
CND00100
COM00010
COM00020
COM00030
COM00040
COM00050
CND00120
CND00130
CND00140
CND00150
CND00160
CND00170
CND00180
CND00190
CND00200
CND00210
CND00220
CND00230
CND00240
CND00250
CND00260
CND00270
CND00280
CND00290
CND00300
CND00310
CND00320
CND00330
CND00340
CND00350
CND00360
CND00370
CND00380
CND00390
CND00400
CND00410
CND00420
CND00430
CND00440
CND00450

FILE: CRDSCN

| | | |
|-----|--|----------|
| | FUNCTION CRDSCN(CARD,GRPDEX,GRPNAM,GROUPS,NOGRP,GRPTR) | CRD00010 |
| C | IMPLICIT INTEGER (A-H,O-Z) | CRD00020 |
| | DIMENSION BUF(4), CARD(62), COMVEC(2), NUMVEC(30) | CRD00030 |
| | DIMENSION GRPDEX(1),GRPNAM(1),GROUPS(1) | CRD00040 |
| | LOGICAL*1 L1(4),L2(32) | CRD00050 |
| | EQUIVALENCE (WRD1,L1(1)),(L2(1),BUF(1)) | CRD00060 |
| C | DATA BLANK/' ', COMMA/',', COMVEC/1.,'/ | CRD00070 |
| C | ----- | CRD00080 |
| C | ----- | CRD00090 |
| C | ----- | CRD00100 |
| C | ----- | CRD00110 |
| | COL = 0 | CRD00120 |
| | J = NXTCHR(CARD,COL) | CRD00130 |
| | IF (J.EQ.BLANK) GO TO 110 | CRD00140 |
| | DO 10 I=1,8 | CRD00150 |
| | J2 = CARD(COL-1+I) | CRD00160 |
| | IF (J2.EQ.COMMA) GO TO 20 | CRD00170 |
| 10 | BUF(I) = J2 | CRD00180 |
| | GO TO 40 | CRD00190 |
| C | 20 DO 30 J=1,8 | CRD00200 |
| | 30 BUF(J) = BLANK | CRD00210 |
| C | 40 N = 1 | CRD00220 |
| | DO 50 I=1,4 | CRD00230 |
| | L1(I) = L2(N) | CRD00240 |
| | N = N + 4 | CRD00250 |
| 50 | CONTINUE | CRD00260 |
| | GRPNAM(NOGRP+1) = WRD1 | CRD00270 |
| | J = FIND12(CARD,COL,COMVEC) | CRD00280 |
| | IF (J.LE.0) GO TO 110 | CRD00290 |
| | J = NUMBER(CARD,COL,NUMVEC,0) | CRD00300 |
| | II = 0 | CRD00310 |
| | DO 90 I=1,J | CRD00320 |
| | JJ = NUMVEC(I) | CRD00330 |
| 80 | II = II+1 | CRD00340 |
| | NUMVEC(II) = JJ | CRD00350 |
| 90 | CONTINUE | CRD00360 |
| | IF (II.LE.0) GO TO 110 | CRD00370 |
| C | NOGRP = NOGRP+1 | CRD00380 |
| | GRPTR = GRPTR +1 | CRD00390 |
| | GRPDEX(NOGRP) = GRPTR | CRD00400 |
| | GROUPS(GRPTR) = II | CRD00410 |
| | DO 100 I=1,II | CRD00420 |
| 100 | GROUPS(GRPTR+I) = NUMVEC(I) | CRD00430 |
| | GRPTR = GRPTR+II | CRD00440 |
| | CRDSCN = 0 | CRD00450 |
| | RETURN | CRD00460 |
| C | | CRD00470 |
| C | | CRD00480 |
| | 110 RETURN | CRD00490 |
| | END | CRD00500 |
| | | CRD00510 |
| | | CRD00520 |
| | | CRD00530 |
| | | CRD00540 |
| | | CRD00550 |

ORIGINAL PAGE IS
OF POOR QUALITY

18-13
344

FILE: DOTDST

```

C      SUBROUTINE DOTDST(MEANS,DOTS,TABLE,TOP)
C      COMPUTES L1 OR L2 DISTANCES AND STORE ON DRUM
C      IMPLICIT INTEGER (A-Z)
C      INCLUDE COMMK1.LIST
C      INCLUDE COMMK4.LIST
C      INCLUDE COMMK6.LIST
C      INCLUDE COMMK15.LIST
C      COMMON/INFORM/NOCLS2,NOSUB2,NOFET2,VARSZ2,TOTVT2,NOFLD2,
C      *      AVAR2,COVAR2,CLSID2,SUBNO2,SURDS2,FLDSV2,VRTX2,
C      *      FETVC2(30),SUBVC2(75),SUHPTR(75),CLSVC2(60),
C      *      KFPPTS(60),NOGRP,GRPNAM(60),GRPDEX(61),
C      *      GRPCHK(61),GROUPS(124)
C      DIMENSION HED1(15),HED2(15),DATE(3),COMENT(15)
C      EQUIVALENCE (HED1(1),HEAD(4)),(DATE(1),HEAD(22)),
C      *      (HED2(1),HEAD(30)),(COMENT(1),HEAD(48))
C      2 COMMON/GLOBAL/HEAD(63),MAPTAP,DATAPE,SAVTAP,BMFILE,BMKEY,
C      *      HISFIL,HISKEY,TRFORM,ERIPTR,ERPKEY,MAPUNT,NOFILE,
C      *      DRUMAD,DRUMDS,PAGSIZ,DATFIL,STAFIL,ASAV,ASAVFL
C      *      ,NHSTUN,NHSTFI,SCTRUN,MAPFIL
C      *      ,DOTUNT,DOTFIL,NCHPAS,TRANSFL,BMTRFL,HISTFL,PCHUNT,
C      *      CRDUNT,PRNTUNT,HANDIO
C      COMMON /LABS/NOCAT,CATNAM(60),NOCL2,CLSNM2(60),NOCAT2,CATNM2(60),
C      *      SUBRAY(120),PTR(60),CATPTR(250),CATDOT(500),
C      *      DOTVEC(250),COND,MIX,PROC,MAPKEY,DOTKEY,STATKY,
C      *      SUNANG,T,NFARST,DIST,NOFEAT,FETVEC(30),OMAPUN,OMAPFI,
C      *      OSAVTP,OSTAFI,NOSUN,ANGLE(A),SIZE,TOTDT2,FLDINF(6),
C      *      CLSSYM(62),STADHS,MEANAD,TABADR,MAPADR,SUNCOR(30),
C      *      ODOTUN,ODOTFI,MANSTA,MANDOT,DSPUNT,DSPFIL,DSPKEY,PRNSTS,
C      *      PRNDOT,FLDNAM,VERTEX(22),NOVRT,NSUN,ANGLES(8)
C      *      ,TOTDT3,FLDAADR,VTXADR
CSEND
C      DIMENSION DOTS(SIZE,1)
C      REAL TABLE(TOTDT2,1)
C      DATA CLSTR/CLST1/,BLANK/0,0/
C      REAL MEANS(NOFET2,1),DSTN,DISTNC(250),SUN,SUNCOR
C      TABADR - DRUM ADDRESS FOR STORING DISTANCE TABLE
C      TAB1 = TABADR
C      RETRIEVE SUN ANGLE CORRECTIONS
C      IF (SUNANG .EQ. 0) GO TO 19
C      SWITCH = 1 - SUNANGLES ARE USER INPUT
C      *      = 0 -SUN ANGLES ARE ON DOT FILE
C      IF (SUNANG .NE. 1) GO TO 18
C      SWITCH = 0
C      CALL SUNFAC(SUNCOR,ANGLE,FETVEC,NOFEAT,SWTCH,DUMMY)
C      GO TO 19
C18 SWITCH = 1
C      CALL SUNFAC(SUNCOR,ANGLES,FETVEC,NOFEAT,SWTCH,DUMMY)
C19 CONTINUE
C      DO 200 I=1,NOSUB2
C      ZERO OUT ARRAY
C      DO 20 IJ=1,TOTDT2
C20 DISTNC(IJ) = 0
C      COMPUTE DISTANCE BETWEEN ALL DOTS FOR EACH CLUSTER
C      DIST = 1 -- L1 DISTANCE
C      *      = 2 -- L2 DISTANCE
C      DO 100 J=1,TOTDT2
C      DSTN = 0
C      DO 50 K=1,NOFET2
C      SUN = SUNCOR(K)
C      DSTN = SUN*DOTS(4+K,J) - SUN*MEANS(K,I)
C      GO TO (30,40),DIST
C30 DISTNC(J) = DISTNC(J) + ABS(DSTN)
C      GO TO 50

```

FILE: DOTDST

```

      40 DISTNC(J) = DISTNC(J) + DSTN**2
      50 CONTINUE
C
    100 IF (DIST .EQ. 2) DISTNC(J) = SORT(DISTNC(J))
      CALL RWRITE(TAB1,DISTNC,TOTDT2,ISTAT)
      TAB1 = TAB1 + TOTDT2
    110 IF (ISTAT .EQ. 1) GO TO 110
    200 CONTINUE
C
      WRITE(6,HEAD)
      WRITE(6,250)
    250 FORMAT(/T50,'CLUSTER-DOT INTER-DISTANCE TABLE')
C
      SUR2 = NOSUR2
      TAB1 = TABADH
      NSUR2 = 1
      NOGRPS = TOP/TOTDT2
      IF (NOGRPS .GT. 15) NOGRPS = 15
    245 IF (NOGRPS .GT. SUR2) NOGRPS = SUR2
      TOTWDS = TOTDT2 * NOGRPS
C
      CALL RREAD(TAB1,TABLE,TOTWDS,ISTAT)
      TSUR2 = NSUR2 + NOGRPS - 1
      WRITE(6,260) (CLSTR ,I=1,NOGRPS)
      WRITE(6,261) (BLANK,SURVC2(K),K=NSUR2,TSUR2)
    260 FORMAT(/78X,15(4X,44))
    261 FORMAT(4X,'DOTS',3X,15(A1,'(',I2,')',3X))
      WRITE(6,262)
    262 FORMAT( )
    265 IF (ISTAT .EQ. 1) GO TO 265
      DO 290 I=1,TOTDT2
      WRITE(6,275)DOTVEC(I),(TABLE(I,J),J=1,NOGRPS)
    275 FORMAT(4X,I3,14X,15(1X,F7.2))
    290 CONTINUE
      TAB1 = TAB1 + TOTWDS
      NSUR2 = TSUR2 + 1
      SUR2 = SUR2 + NOGRPS
      IF (SUR2 .GT. 0) GO TO 245
C
      RETURN
      END

```

DOT00800
 DOT00810
 DOT00820
 DOT00830
 DOT00840
 DOT00850
 DOT00860
 DOT00870
 DOT00880
 DOT00890
 DOT00900
 DOT00910
 DOT00920
 DOT00930
 DOT00940
 DOT00950
 DOT00960
 DOT00970
 DOT00980
 DOT00990
 DOT01000
 DOT01010
 DOT01020
 DOT01030
 DOT01040
 DOT01050
 DOT01060
 DOT01070
 DOT01080
 DOT01090
 DOT01100
 DOT01110
 DOT01120
 DOT01130
 DOT01140
 DOT01150
 DOT01160
 DOT01170
 DOT01180
 DOT01190
 DOT01200

ORIGINAL PAGE
 OF POOR QUALITY

FILE: DSPTAP

```

SURROUTINE DSPTAP(SURNO,SURDES,FLDSAV,VERTX,CATVEC,SUBVEC,MEANS, DSP00010
* COVAR, TOP, DATA, NOFLD, TOTVRT) DSP00020
IMPLICIT INTEGER (A-Z) DSP00030
INCLUDE COMK1.LIST DSP00040
INCLUDE COMK4.LIST DSP00050
INCLUDE COMK6.LIST DSP00060
INCLUDE CMK15.LIST DSP00070
COMMON/INFORM/NOCLS2,NOSUR2,NOFET2,VARSZ2,TOTVT2,NOFLD2, DSP00080
* AVAR2,COVAR2,CLSIN2,SURNO2,SURDS2,FLDSV2,VERTX2, DSP00090
* FETVC2(30),SURVC2(75),SUBPTR(75),CLSV2(60), DSP00100
* KFPPTS(60),NOGRP,GRPNAM(60),GRPDEX(61), DSP00110
* GRPCHK(61),GROUPS(124) DSP00120
DIMENSION HED1(15),HED2(15),DATE(3),COMENT(15) DSP00130
EQUIVALENCE (HED1(1),HEAD(4)),(DATE(1),HEAD(22)), DSP00140
2 (HED2(1),HEAD(30)),(COMENT(1),HEAD(48)) DSP00150
COMMON/GLOBAL/HEAD(63),MAPTAP,DATAPE,SAVTAP,9MFILE,9MKEY, DSP00160
* HISFIL,HISKEY,TRFORM,ERIPTR,ERPKEY,MAPUNT,NOFILE, DSP00170
* DRUMAD,DRMWD5,PAGSIZ,DATAFIL,STAFIL,ASAV,ASAVFL DSP00180
* ,NHSTIN,NHSTFI,SCTRIIN,MAPFIL DSP00190
* ,DOTUNT,DOTFIL,NCHPAS,TRANSFL,BMTRFL,HISTFL,PCHUNT, DSP00200
* CROUT,PRUNT,RANDIO DSP00210
COMMON /LABS/NOCAT,CATNAM(60),NOCL2,CLSNM2(60),NOCAT2,CATNM2(60), DSP00220
* SURPAY(120),PTR(60),CATPTR(250),CATDOT(500), DSP00230
* DOTVEC(250),COND,MIX,PROC,MAPKEY,DOTKEY,STATKY, DSP00240
* SUNANG,T,NEARST,DIST,NOFEAT,FETVEC(30),UMAPUN,OMAPFI, DSP00250
* OSAVTP,OSTAFI,NOSUN,ANGLE(A),SIZE,TOTDT2,FLDINF(6), DSP00260
* CLSSYM(62),STADRS,MEANAD,TABADR,MAPADR,SUNCOR(30), DSP00270
* ODOTUN,ODOTFI,MANSTA,MANDOT,DSPUNT,DSPFIL,DSPKEY,PRNSTS, DSP00280
* PRNDOT,FLDNAM,VERTEX(22),NOVRT,NSUN,ANGLES(8) DSP00290
* ,TOTDT3,FLDADR,VTXADR DSP00300
C*END DSP00310
DIMENSION SURNO(1),SURDES(1),FLDSAV(4,1),VERTX(2,1),CATVEC(1) DSP00320
DIMENSION SUBVEC(1),DATA(1),IP(1000),NEWSUB(62) DSP00330
DIMENSION CLSV2(60),DUMMY(100) DSP00340
REAL MEANS(NOFET2,1),COVAR(VARSZ2,1),VR(1000) DSP00350
C POSITION TAPE DSP00360
C DSP00370
C CALL FSBSFL(DSPUNT,DSPFIL,IST) . DSP00380
C DSP00390
C RUN HEADER RECORD NO. 1 DSP00400
C DSP00410
C DSP00420
DO 10 I=1,100 DSP00430
10 DUMMY(I)=1 DSP00440
NOFLD2 = NOFLD DSP00450
TOTVT2 = TOTVRT DSP00460
NCAT = 0 DSP00470
C**** JULY 12 1978 NOCAT USED INSTEAD OF NCAT IN MAPTAP FIRST RECORD DSP00480
C DSP00490
WRITE(DSPUNT)((DATE(I),I=1,2),(DUMMY(I),I=1,3),NOCAT,NOFLD2,NOSUB2, DSP00500
* NOFET2,TOTVT2,NOCAT,VARSZ2,(FETVC2(I),I=1,NOFET2) DSP00510
C DSP00520
K = 0 DSP00530
DO 40 I=1,NOCAT DSP00540
III = SURNO(I) DSP00550
DO 40 J=1,III DSP00560
K = K + 1 DSP00570
40 CLSVEC(K) = 1 DSP00580
C RUN HEADER RECORD NO. 2 DSP00590
WRITE(DSPUNT)(CATNAM(I),I=1,NOCAT),(CATNAM(I),I=1,NOCAT), DSP00600
* (SURNO(I),I=1,NOCAT),(SURDES DSP00610
* (I),I=1,NOSUR2),((FLDSAV(I,J),I=1,4),J=1,NOFLD2),((VERTX(I,J) DSP00620
* ,I=1,2),J=1,TOTVT2),(CLSVEC(I),I=1,NOSUB2),(CLSVEC(I),I=1,NOSUB2) DSP00630
* ,(DUMMY(I),I=1,NOCAT),(KFPPTS(I),I=1,NOSUR2) DSP00640
C DSP00650
C RUN HEADER RECORD NO. 3 DSP00660
C DSP00670
C DSP00680
MEAN1 = STADRS + VARSZ2 * NOSUB2 DSP00690
DO 100 J=1,NOSUB2 DSP00700
KK = SURVEC(J) DSP00710
MEANS2 = MEAN1 + NOFET2*(KK-1) DSP00720
COVAR1 = STADRS + VARSZ2*(KK-1) DSP00730
CALL RPFAD(COVAR1,COVAR(1,J),VARSZ2,ISTAT) DSP00740
50 IF (ISTAT.EQ. 1) GO TO 50 DSP00750
CALL RPFAD(MEANS2,MEANS(1,J),NOFET2,ISTAT1) DSP00760
60 IF (ISTAT1.EQ. 1) GO TO 60 DSP00770
C DSP00780
100 CONTINUE DSP00790
C

```

FILE: DSPTAP

```

      WRITE(DSPUNT)((COVAR(I,J),I=1,VARSZ2),J=1,NOSUB2),((MEANS(I,J),
      * I=1,NOFET2),J=1,NOSUB2)
C
C
C
      RUN HEADER RECORD NO. 4
C
C
      WRITE(DSPUNT)((COVAR(I,J),I=1,VARSZ2),J=1,NOSUB2),
      * (DUMMY(I),I=1,NOSUB2),(DUMMY(I),I=1,NOSUB2)
C
C
      FIELD RECORD
C
      LINSTR = FLDINF(1)
      LINEND = FLDINF(2)
      LININC = FLDINF(3)
      SAMSTR = FLDINF(4)
      SAMEND = FLDINF(5)
      SAMINC = FLDINF(6)
      PTS = (SAMEND-SAMSTR)/SAMINC + 1
      LINES = (LINEND-LINSTR)/LININC + 1
C
      WRITE(DSPUNT)(FLDINF(I),I=1,6),PTS,LINES,FLDNAM,NOVRT,
      * (VERTEX(I),I=1,NOVRT),(VERTEX(I+NOVRT),I=1,NOVRT)
C
C
C
      NEWSUB -- NEW SUBCLASS NUMBERS
C
      DO 120 I=1,NOSUB2
      K = SURVEC(I)
120  NEWSUR(K) = I
      DO 130 I=1,PTS
      VR(I) = 0.0
130  MAP = DNUMAD
      ILINE = LINES
      NOLINE = TOP/PTS
135  IF (NOLINE .GT. ILINE) NOLINE = ILINE
      TOTPIX = NOLINE * PTS
      CALL WREAD(MAP,DATA,TOTPIX,ISTAT2)
      MAP = MAP + TOTPIX
137  IF (ISTAT2 .EQ. 1) GO TO 137
      DO 150 I=1,NOLINE
      II = II + 1
      N = LINSTR + LININC*(II-1)
C
      DO 140 J=1,PTS
      IDUM=(I-1)*PTS+J
      JJ = DATA(IDUM)
140  IR(J) = NEWSUR(JJ)
C
      WRITE(DSPUNT)N,((IR(K),K=1,PTS),(VR(K),K=1,PTS)
150  CONTINUE
      ILINE = ILINE - NOLINE
      IF (ILINE .LE. 0) GO TO 155
      GO TO 135
155  N = 0
      WRITE(DSPUNT)N,((IR(I),I=1,PTS),(VR(I),I=1,PTS)
      PTS = 0
      WRITE(DSPUNT)(FLDINF(I),I=1,6),PTS,LINES,FLDNAM,NOVRT,
      * (VERTEX(I),I=1,NOVRT),(VERTEX(I+NOVRT),I=1,NOVRT)
      ENDFILE DSPUNT
C
      RETURN
      END

```

DSP00800
 DSP00810
 DSP00820
 DSP00830
 DSP00840
 DSP00850
 DSP00860
 DSP00870
 DSP00880
 DSP00890
 DSP00900
 DSP00910
 DSP00920
 DSP00930
 DSP00940
 DSP00950
 DSP00960
 DSP00970
 DSP00980
 DSP00990
 DSP01000
 DSP01010
 DSP01020
 DSP01030
 DSP01040
 DSP01050
 DSP01060
 DSP01070
 DSP01080
 DSP01090
 DSP01100
 DSP01110
 DSP01120
 DSP01130
 DSP01140
 DSP01150
 DSP01160
 DSP01170
 DSP01180
 DSP01190
 DSP01200
 DSP01210
 DSP01220
 DSP01230
 DSP01240
 DSP01250
 DSP01260
 DSP01270
 DSP01280
 DSP01290
 DSP01300
 DSP01310
 DSP01320
 DSP01330
 DSP01340
 DSP01350
 DSP01360
 DSP01370
 DSP01380
 DSP01390
 DSP01400

ORIGINAL PAGE IS
OF POOR QUALITY

FILE: FILERO

```

SUBROUTINE FILERO (ARRAY, TOP, NOFLD, TOTVRT, FLDSAV, VERTX)
  READS IN ALL NEEDED FILES

  DRUM ADDRESSES
  DRUMAD - BEGINNING ADDRESS FOR MAPFIL
  STADRS - BEGINNING ADDRESS FOR COVAR AND MEANS
  TABADR - BEGINNING ADDRESS FOR DISTANCE TABLE
  MAPADR - BEGINNING ADDRESS FOR NSAMP-110 PTS OF COND. OR
           MIXED CLUSTER MAP
  FLNADR - BEGINNING ADDRESS FOR FIELD INFO
  VTXADR - BEGINNING ADDRESS FOR VERTICES
  COVAR2 - BEGINNING ADDRESS FOR DOT DATA

  IMPLICIT INTEGER (A-Z)
  LIMIT = 5000
  INCLUDE COMK1.LIST
  INCLUDE COMK6.LIST
  INCLUDE COMK15.LIST
  COMMON/INFORM/NOCLS2, NOSUR2, NOFET2, VARSZ2, TOTVT2, NOFLD2,
  * AVAR2, COVAR2, CLSID2, SURNO2, SUROS2, FLDSV2, VERTX2,
  * FETVC2(30), SURVC2(75), SUPTR(75), CLSYC2(60),
  * KFPPTS(60), NOGRP, GRPNAM(60), GRPDEX(61),
  * GRPCHK(61), GROUPS(124)
  COMMON/GLOBAL/HEAD(63), MAPTAP, DATAPE, SAVTAP, BMFILE, BMKEY,
  * HISFIL, HISKEY, TRFORM, ERIPTP, ERPKEY, MAPUNT, NOFILE,
  * DRUMAD, DRUMDS, PAGESZ, DATFIL, STAFIL, ASAV, ASAVFL
  * NHSTUN, NHSTFI, SCTRUN, MAPFIL
  * DOTUNT, DOTFIL, NCHPAS, TRNSFL, BMTRFL, HISTFL, PCHUNT,
  * CRDUNT, PRTUNT, WANDIO
  COMMON /LAHS/NOCAT, CATNAM(60), NOCL2, CLSNM2(60), NOCAT2, CATNM2(60),
  * SUHRA(120), PTR(60), CATPTR(250), CATDOT(500),
  * DOTVEC(250), COND, MIX, PROC, MAPKEY, DOTKEY, STATKY,
  * SUNANG, T, NEARST, DIST, NOFFAT, FETVEC(30), OMAPUN, OMAFFI,
  * OSAVTP, OSTAFL, NOSUN, ANGLE(8), SIZE, TOTDT2, FLDINF(6),
  * CLSSYM(62), STADRS, MEANAD, TABADR, MAPADR, SUNCOR(30),
  * ODOTUN, ODOTFI, MANSTA, MANDOT, DSPUNT, DSPFIL, DSPKEY, PRNSTS,
  * PRNDOT, FLNDR, VERTEX(22), NOVRT, NSUN, ANGLES(8)
  * TOTDT3, FLNADR, VTXADR

  CSEND
  DIMENSION ARRAY(1)
  DIMENSION FLDSAV(4,1), VERTX (2,1)
  DIMENSION FETVC3(30)

  COVAR2 = 1

  READ IN MAPFIL AND STORE ON THE DRUM
  IF (MAPKEY .EQ. 0) GO TO 100
  I = LAREAD(FLDNAM, VERTEX, FLDINF, NOVRT)
  NOLINE = (FLDINF(2) - FLDINF(1))/FLDINF(3) + 1
  NSAMP = (FLDINF(5) - FLDINF(4))/FLDINF(6) + 1
  TOTPIX = NOLINE*NSAMP

  CALL STOMAP(NOLINE, NSAMP, ARRAY, TOP, DRUMAD)
  READ IN STAT FILE
  100 IF (STATKY .EQ. 0) GO TO 200
  CALL REDSAV(ARRAY, TOP, HMFLG)
  STASIZ = (VARSZ2 + NOFET2) * NOSUB2
  STADRS = DRUMAD + TOTPIX
  CALL RWRITE(STADRS, ARRAY(COVAR2), STASIZ, ISTAT)
  110 IF (ISTAT .EQ. 1) GO TO 110
  READ IN DOTFIL
  200 IF (DOTKEY .EQ. 0) RETURN
  TOTSTO = TOP - COVAR2
  IF (TOTSTO .GE. LIMIT) GO TO 220

```

FIL00010
 FIL00020
 FIL00030
 FIL00040
 FIL00050
 FIL00060
 FIL00070
 FIL00080
 FIL00090
 FIL00100
 FIL00110
 FIL00120
 FIL00130
 FIL00140
 FIL00150
 FIL00160
 FIL00170
 FIL00180
 FIL00190
 FIL00200
 FIL00210
 FIL00220
 FIL00230
 FIL00240
 FIL00250
 FIL00260
 FIL00270
 FIL00280
 FIL00290
 FIL00300
 FIL00310
 FIL00320
 FIL00330
 FIL00340
 FIL00350
 FIL00360
 FIL00370
 FIL00380
 FIL00390
 FIL00400
 FIL00410
 FIL00420
 FIL00430
 FIL00440
 FIL00450
 FIL00460
 FIL00470
 FIL00480
 FIL00490
 FIL00500
 FIL00510
 FIL00520
 FIL00530
 FIL00540
 FIL00550
 FIL00560
 FIL00570
 FIL00580
 FIL00590
 FIL00600
 FIL00610
 FIL00620
 FIL00630
 FIL00640
 FIL00650
 FIL00660
 FIL00670
 FIL00680
 FIL00690
 FIL00700
 FIL00710
 FIL00720
 FIL00730
 FIL00740
 FIL00750
 FIL00760
 FIL00770
 FIL00780
 FIL00790

FILE: FILERD

| | | |
|-----|---|----------|
| | WRITE(6,210) | FIL00800 |
| 210 | FORMAT(' NOT ENOUGH CORE TO STORE DOTFIL') | FIL00810 |
| | CALL CMERK | FIL00820 |
| C | | FIL00830 |
| 220 | TYPSWT = 3 | FIL00840 |
| | CALL RDDOTS(ARRAY(COVAR2),DOTVEC,TOTDT3,TYPSWT,SIZE,TOTDT2,NOCAT, | FIL00850 |
| | 1 CATNAM,NOFEAT,FETVEC,NOFET3,FETVC3,NOSUN,ANGLE,NOFLD, | FIL00860 |
| | 2 TOTVRT,FLDSAV,VERTX,DUMMY) | FIL00870 |
| | CALL WRTFLD(FLDSAV,VERTX,NOFLD,2,CATNAM,DUMMY) | FIL00880 |
| | TARADR = STADRS + (VARSZ2*NOFEAT)*NOSUB2 | FIL00890 |
| | MAPADR = TARADR + NOSUB2*TOTDT2 | FIL00900 |
| | TOTAL = MAPADR | FIL00910 |
| | FLDADR = TOTAL | FIL00920 |
| | IF (MIX.EQ.0 .AND. COND.EQ.0) GO TO 222 | FIL00930 |
| | FLDADR = TOTAL + (NSAMP-110)*NOLINE | FIL00940 |
| | IF (FLDADR.LT. TOTAL) FLDADR = TOTAL | FIL00950 |
| | TOTAL = FLDADR | FIL00960 |
| 222 | IF (DSPKEY.EQ.0) GO TO 225 | FIL00970 |
| | VTXADR = TOTAL + 4*NOFLD | FIL00980 |
| | TOTAL = VTXADR + TOTVRT*2 | FIL00990 |
| C | | FIL01000 |
| 225 | CONTINUE | FIL01010 |
| | IF (TOTAL.LE. (DRUMAD+DRMWDS)) GO TO 230 | FIL01020 |
| | WRITE(6,400) | FIL01030 |
| 400 | FORMAT(' NOT ENOUGH CORE DRUM SPACE OF CLUSTER MAP INFO') | FIL01040 |
| 230 | CONTINUE | FIL01050 |
| C | | FIL01060 |
| C | | FIL01070 |
| C | RETURN | FIL01080 |
| C | | FIL01090 |
| | END | FIL01100 |

FILE KNEAR

```

C      SURROUTINE KNEAR(DOTS,SUBVEC,SUBNO,CATVEC,ITER,TAB1,SWTCH,
C      • CATNUM,CLUNUM,MEANS,DOTSUM)
C      LABELS BY THE K-NEAREST NEIGHBOR PROCEDURE
C      IMPLICIT INTEGER (A-Z)
C      REAL DISTNC(250)
C      DIMENSION DOTS(SIZE,1),SUBNO(1),SUBVEC(60)
C      DIMENSION CATVEC(1),DOTNAM(250),DOTSUM(60,60)
C      DIMENSION CATGRY(60),DOTCLU(250)
C      INCLUDE COMRK1,LIST
C      INCLUDE COMRK4,LIST
C      INCLUDE COMRK15,LIST
C      INCLUDE COMRK6,LIST
C      COMMON/INFORM/NOCLS2,NOSUB2,NOFET2,VARSZ2,TOTVT2,NOFLD2,
C      • AVAR2,COVAR2,CLS1D2,SURNO2,SUBDS2,FLDSV2,VERTX2,
C      • FETVC2(30),SURVC2(75),SURPTR(75),CLSV2(60),
C      • KEPPTS(60),NOGRP,GRPNAM(60),GRPDEX(61),
C      • GRPCHK(61),GROUPS(124)
C      DIMENSION HED1(15),HED2(15),DATE(3),COMENT(15)
C      EQUIVALENCE (HED1(1),HEAD(4)),(DATE(1),HEAD(22)),
C      • (HED2(1),HEAD(30)),(COMENT(1),HEAD(48))
C      2 COMMON /LABS/NOCAT,CATNAM(60),NOCL2,CLSNM2(60),NOCAT2,CATNM2(60),
C      • SURRAY(120),PTR(60),CATPTR(250),CATDOT(500),
C      • DOTVEC(250),COND,MIX,PROC,MAPKEY,DOTKEY,STATKY,
C      • SUNANG,T,NEARST,DIST,NOFEAT,FETVEC(30),OMAPUN,OMAPFI,
C      • OSAVTP,OSTAFI,NOSUN,ANGLE(8),SIZE,TOTDT2,FLDINF(6),
C      • CLSSYM(62),STADRS,MEANAD,TABADR,MAPADR,SUNCOR(30),
C      • ODOTUN,ODOTFI,MANSTA,MANDOT,DSPUNT,DSPFIL,DSPKEY,PRNSTS,
C      • PRNDOT,FLDNAM,VERTEX(22),NOVRT,NSUN,ANGLES(8)
C      • TOTDT3,FLDACR,VTXADR
C      COMMON/GLOBAL/HEAD(63),MAPTAP,DATAP,SAVTAP,BMFILE,BMKEY,
C      • HISFIL,HISKEY,TRFORM,ERIPTP,ERPKEY,MAPUNT,NOFILE,
C      • DRUMAD,DRMWS,PAGSIZ,DATFIL,STAFIL,ASAV,ASAVFL
C      • NHSTUN,NHSTFI,SCRUN,MAPFIL
C      • DOTUNT,DOTFIL,NCHPAS,TRNSFL,BMTRFL,HISTFL,PEHUNT,
C      • CRDUNT,PRUNT,RANDIO
C$END
C      NO OF DOTS TO COMPARE
C      REAL MEANS(NOFET2,1)
C      DIMENSION DOTVC2(250),TIES(250)
C      SAVE DOTVEC
C      DO 5 I=1,TOTDT2
C      5 DOTVC2(I) = DOTVEC(I)
C      FIND CLUSTER NUMBER OF DOTS IF MAP AVAILABLE
C      IF (MAPKEY.EQ.0)GO TO 15
C      NSAMP=(FLDINF(5)-FLDINF(4))/FLDINF(6)+1
C      DO 12 I=1,TOTDT2
C      ILINE=DOTS(2,I)
C      ISAMP=DOTS(1,I)
C      PIXADR=DRUMAD+(ILINE/FLDINF(3)-1)*NSAMP+ISAMP/FLDINF(6)-1
C      CALL RREAD(PIXADR,NUMBER,1,ISTAT)
C      DOTCLU(I)=NUMBER
C      12 CONTINUE
C      15 IF (SWTCH.EQ.1)GO TO 6
C      IF (SWTCH.EQ.0) WRITE(6,HEAD)
C      WRITE(6,10)NEARST
C      10 FORMAT(/T43,'LABELING BY',I3,'-NEAREST NEIGHBOR PROCEDURE'/)
C      KNGHRR = NEARST
C      WRITE(6,1111)
C      1111 FORMAT(20X,'CLUSTER LABELING DETAILS',/)
C      WRITE(6,1000)
C      1000 FORMAT(3X,'CLUSTER',2X,'CLUSTER',3X,'DOT',4X,'DOT',
C      1 7X,'DOT',6X,'DOT',/,3X,'NUMBER',4X,'LABEL',3X,'LABEL',
C      2 2X,'NUMBER',2X,'DISTANCE',2X,'CLUSTER',/)
C      DO 500 I=1,ITER

```

KNE00010
 KNE00020
 KNE00030
 KNE00040
 KNE00050
 KNE00060
 KNE00070
 KNE00080
 KNE00090
 KNE00100
 KNE00110
 KNE00120
 KNE00130
 KNE00140
 KNE00150
 KNE00160
 KNE00170
 KNE00180
 KNE00190
 KNE00200
 KNE00210
 KNE00220
 KNE00230
 KNE00240
 KNE00250
 KNE00260
 KNE00270
 KNE00280
 KNE00290
 KNE00300
 KNE00310
 KNE00320
 KNE00330
 KNE00340
 KNE00350
 KNE00360
 KNE00370
 KNE00380
 KNE00390
 KNE00400
 KNE00410
 KNE00420
 KNE00430
 KNE00440
 KNE00450
 KNE00460
 KNE00470
 KNE00480
 KNE00490
 KNE00500
 KNE00510
 KNE00520
 KNE00530
 KNE00540
 KNE00550
 KNE00560
 KNE00570
 KNE00580
 KNE00590
 KNE00600
 KNE00610
 KNE00620
 KNE00630
 KNE00640
 KNE00650
 KNE00660
 KNE00670
 KNE00680
 KNE00690
 KNE00700
 KNE00710
 KNE00720
 KNE00730
 KNE00740
 KNE00750
 KNE00760

FILE KNEAR

```

C      READ IN DISTANCES FROM DRUM ONE CLUSTER AT A TIME
C      TIE = 0
C      II = 1
C      CALL RREAD(TAB1,DISTNC,TOTDT2,ISTAT)
50  IF (ISTAT.EQ. 1) GO TO 50
    TAB1 = TAB1 + TOTDT2
C      SORT DISTANCES IN ASCENDING ORDER
C      DO 55 J=1,TOTDT2
    DOTVEC(J) = DOTVC2(J)
55  DOTNAM(J) = DOT5(4,J)
    CALL ASCEND(DISTNC,TOTDT2,DOTNAM,DOTVEC)
C      REINITIALIZE ARRAYS
57  DO 60 J=1,NOCAT
60  CATGRY(J) = 0
    MAX = 0
    DO 70 JJ=1,KNGHBR
    L = DOTNAM(JJ)
C      RETRIEVE CATEGORY NO.
C      CATGRY(L) = CATGRY(L) + 1
    IF (CATGRY(L) .LE. MAX) GO TO 70
    MAX = CATGRY(L)
    CATNUM = L
70  CONTINUE
C      CHECK FOR A TIE
C      IF (KNGHBR .EQ. 1) GO TO 100
C      DO 80 III=1,NOCAT
    IF (III .EQ. CATNUM) GO TO 80
    IF (IMAX .EQ. CATGRY(III)) GO TO 90
80  CONTINUE
C      NO TIES OCCURRED
C      GO TO 100
C      A TIE OCCURRED - DECREASE K-DOTS BY 1 AND REPEAT PROCESS
90  KNGHRR = KNGHBR - 1
    TIE = TIE + 1
    TIES(TIE) = KNGHRR + 1
    GO TO 57
C      ASSIGN CLUSTER TO CATEGORY
100 IF (SWTCH .EQ. 1) II = CLUNUM
    CATVEC(II) = CATNUM
C      PRINT CLUSTER INFORMATION
C      WRITE(6,1100) II,CATNAM(CATNUM)
1100 FORMAT(/,5X,12,8X,1A4)
    DO 110 J=1,KNGHBR
    K=DOTNAM(J)
    L=DOTCLU(DOTVEC(J))
    IF (J.EQ.1) WRITE(6,1201) CATNAM(K),DOTVEC(J),DISTNC(J)
1201 FORMAT(1H+.T24,1A4,2X,13,4X,F7.2)
    IF (J.GT.1) WRITE(6,1200) CATNAM(K),DOTVEC(J),DISTNC(J)
1200 FORMAT(23X,1A4,2X,13,4X,F7.2)
    IF (DOTKEY.EQ.1) WRITE(6,1210) L
1210 FORMAT(1H+.T4R,12)
    DOTSUM(II,K)=DOTSUM(II,K)+1
110  CONTINUE
    IF (TIE .EQ. 0) GO TO 490
    WRITE(6,185)
185  FORMAT(/)
    WRITE(6,190)
190  FORMAT(23X,'A TIE OCCURRED.',3X,'THE FOLLOWING DOT(S) WERE DISCARDKNE01520

```

KNE00770
 KNE00780
 KNE00790
 KNE00800
 KNE00810
 KNE00820
 KNE00830
 KNE00840
 KNE00850
 KNE00860
 KNE00870
 KNE00880
 KNE00890
 KNE00900
 KNE00910
 KNE00920
 KNE00930
 KNE00940
 KNE00950
 KNE00960
 KNE00970
 KNE00980
 KNE00990
 KNE01000
 KNE01010
 KNE01020
 KNE01030
 KNE01040
 KNE01050
 KNE01060
 KNE01070
 KNE01080
 KNE01090
 KNE01100
 KNE01110
 KNE01120
 KNE01130
 KNE01140
 KNE01150
 KNE01160
 KNE01170
 KNE01180
 KNE01190
 KNE01200
 KNE01210
 KNE01220
 KNE01230
 KNE01240
 KNE01250
 KNE01260
 KNE01270
 KNE01280
 KNE01290
 KNE01300
 KNE01310
 KNE01320
 KNE01330
 KNE01340
 KNE01350
 KNE01360
 KNE01370
 KNE01380
 KNE01390
 KNE01400
 KNE01410
 KNE01420
 KNE01430
 KNE01440
 KNE01450
 KNE01460
 KNE01470
 KNE01480
 KNE01490
 KNE01500
 KNE01510
 KNE01520

FILE KNEAR

```

      *ED*/)
      DO 200 JJ=1,TIE
      J = TIES(JJ)
      K = DOTNAM(J)
      L=DOTCLU(DOTVEC(J))
      WRITE(6,1200)CATNAM(K),DOTVEC(J),DISTNC(J)
      IF (DOTKEY.EQ.1)WRITE(6,1210)L
200  CONTINUE
      KNGHBR=KNGHBR+TIE
C
490  CONTINUE
      IF (SWTCH .NE. 1) GO TO 500
C
      RESTORE DOTVEC
C
      DO 210 J=1,TOTDT2
210  DOTVEC(J) = DOTVC2(J)
500  CONTINUE
      IF (SWTCH .EQ. 1) RETURN
C
      WRITE DOT SUMMARY
C
      WRITE(6,2222)
      FORMAT(1H,20X,'CLUSTER LABELING SUMMARY',/)
      WRITE(6,1300)
1300  FORMAT(3X,'CLUSTER',20X,'NUMBER OF DOTS USED (BY CATEGORY NAME)',)
      ISTRT=1
      IEND=NOCAT
      IF (IEND.GT.15) IEND=15
      WRITE(6,1305)
1305  FORMAT(3X,'NUMBER',3X,'LABEL',7X,50(1H-))
      WRITE(6,1330)
1330  FORMAT(1H,T20,'TOTAL')
      WRITE(6,1310) (CATNAM(IJ),IJ=1,IEND)
1310  FORMAT(30X,1A4,14(3X,1A4))
      DO 600 I=1,ITER
      TOTAL=0
      DO 550 J=1,NOCAT
      TOTAL=TOTAL+DOTSUM(I,J)
550  K=CATVEC(I)
      WRITE(6,1320) I,CATNAM(K),TOTAL,(DOTSUM(I,J),J=1,IEND)
1320  FORMAT(//,5X,12,6X,1A4,15,2X,15(2X,15))
600  CONTINUE
602  IF (IEND.EQ.NOCAT)GO TO 650
      ISTRT=IEND+1
      IEND=NOCAT
      IF (IEND.GT.ISTRT+14) IEND=ISTRT+14
      WRITE(6,1350)
1350  FORMAT(//)
      WRITE(6,1300)
      WRITE(6,1305)
      WRITE(6,1310) (CATNAM(IJ),IJ=ISTRT,IEND)
      DO 610 I=1,ITER
      K=CATVEC(I)
      WRITE(6,1340) I,CATNAM(K),(DOTSUM(I,J),J=ISTRT,IEND)
1340  FORMAT(//,5X,12,6X,1A4,7X,15(2X,15))
610  CONTINUE
      GO TO 602
C
      GROUP LABELED CLUSTER ACCORDING TO CATEGORY
C
650  K = 0
      DO 510 I=1,NOCAT
      DO 510 J=1,NOSUB2
C
      IF (CATVEC(J) .NE. I) GO TO 510
      SUBNO(I) = SUBNO(I) + 1
      K = K + 1
      SUBVEC(K) = J
C
510  CONTINUE
C
      RETURN
      END

```

KNE01530
 KNE01540
 KNE01550
 KNE01560
 KNE01570
 KNE01580
 KNE01590
 KNE01600
 KNE01610
 KNE01620
 KNE01630
 KNE01640
 KNE01650
 KNE01660
 KNE01670
 KNE01680
 KNE01690
 KNE01700
 KNE01710
 KNE01720
 KNE01730
 KNE01740
 KNE01750
 KNE01760
 KNE01770
 KNE01780
 KNE01790
 KNE01800
 KNE01810
 KNE01820
 KNE01830
 KNE01840
 KNE01850
 KNE01860
 KNE01870
 KNE01880
 KNE01890
 KNE01900
 KNE01910
 KNE01920
 KNE01930
 KNE01940
 KNE01950
 KNE01960
 KNE01970
 KNE01980
 KNE01990
 KNE02000
 KNE02010
 KNE02020
 KNE02030
 KNE02040
 KNE02050
 KNE02060
 KNE02070
 KNE02080
 KNE02090
 KNE02100
 KNE02110
 KNE02120
 KNE02130
 KNE02140
 KNE02150
 KNE02160
 KNE02170
 KNE02180
 KNE02190
 KNE02200
 KNE02210
 KNE02220
 KNE02230
 KNE02240
 KNE02250
 KNE02260

FILE: LABDOT

```

C      SUBROUTINE LABDOT(DOTS)
C      LABDOT UPDATES DOTFIL
C      IMPLICIT INTEGER (A-Z)
C      INCLUDE CMRK15.LIST
COMMON /LABS/NOCAT,CATNAM(60),NUCL2,CLSNM2(60),NOCAT2,CATNM2(60),
*      SUBRAY(120),PTR(60),CATPTR(250),CATDOT(500),
*      DOTVEC(250),COND,MIX,PHOC,MAPKEY,DOTKEY,STATKY,
*      SUNANG,T,NEAPST,NIST,NOFEAT,FETVEC(30),OMAPUN,OMAPFI,
*      OSAVTP,OSTAFI,NOSUN,ANGLE(8),SIZE,TOTDT2,FLOINF(8),
*      CLSSYM(62),STADRS,MEANAD,TAHADP,MAPADR,SUNCOR(30),
*      ODOTUN,ODOTFI,MANSTA,MANDOT,DSPUNT,DSPFIL,DSPKEY,PRNSTS,
*      PRNDOT,FLODNAM,VERTEX(22),NOVHT,NSUN,ANGLES(8)
*      ,TOTDT3,FLOADR,VTXADR
C$END
      DIMENSION DOTS(SIZE,1),CATNO(60)
      CHECK CATEGORY NAMES FOR NEW ENTRIES
      DO 100 I=1,NOCAT2
      DO 90 J=1,NOCAT
      IF (CATNM2(I).EQ. CATNAM(J))GO TO 95
90  CONTINUE
      NOCAT = NOCAT + 1
      INSERT NEW CATEGORY
      CATNAM(NOCAT) = CATNM2(I)
      CATNO(I) = NOCAT
      GO TO 100
      95 CATNO(I) = J
      100 CONTINUE
      DO 150 I=1,NOCAT2
      RETRIEVE BEGIN. AND END. POINTER
      IB = CATPTR(I) + 1
      IE = IB + CATDOT(IB-1) - 1
      DO 120 J=IB,IE
      RETRIEVE DOT NO AND CHANGE CATEGORY NO FOR DOT
      K = CATDOT(J)
      120 DOTS(4,K) = CATNO(I)
      150 CONTINUE
      RETURN
      END

```

LAB00010
 LAB00020
 LAB00030
 LAB00040
 LAB00050
 LAB00060
 LAB00070
 LAB00080
 LAB00090
 LAB00100
 LAB00110
 LAB00120
 LAB00130
 LAB00140
 LAB00150
 LAB00160
 LAB00170
 LAB00180
 LAB00190
 LAB00200
 LAB00210
 LAB00220
 LAB00230
 LAB00240
 LAB00250
 LAB00260
 LAB00270
 LAB00280
 LAB00290
 LAB00300
 LAB00310
 LAB00320
 LAB00330
 LAB00340
 LAB00350
 LAB00360
 LAB00370
 LAB00380
 LAB00390
 LAB00400
 LAB00410
 LAB00420
 LAB00430
 LAB00440
 LAB00450
 LAB00460
 LAB00470
 LAB00480
 LAB00490
 LAB00500
 LAB00510
 LAB00520
 LAB00530
 LAB00540
 LAB00550

ORIGINAL PAGE IS
OF POOR QUALITY

18-23
354

FILE LABLR

```

C      SURROUTINE LABLR (ARRAY, TOP, NOFLD, TOTVRT, FLDSAV, VERTX, MEANS, EXITT)
C      IMPLICIT INTEGER (A-Z)
C      LIMIT = 3135
C      INCLUDE COMBK1.LIST
C      INCLUDE COMBK6.LIST
C      INCLUDE CMHKS.LIST
COMMON /INFORM/ NOCLS2, NOSUR2, NOFET2, VARSZ2, TOTVT2, NOFLD2,
•      AVAR2, COVAR2, CLSID2, SUBNO2, SUBDS2, FLOSV2, VERTX2,
•      FETVC2(30), SURVC2(75), SURPTR(75), CLSVC2(60),
•      KPPPTS(60), NOGRP, GRPNAM(60), GRPDEX(61),
•      GRPCHK(61), GROUPS(124)
COMMON /GLOBAL/ HEAD(63), MAPTAP, DATAPF, SAVTAP, BMFILE, BMKEY,
•      HISFIL, HISKEY, TRFOHM, ERIPTP, ERPKEY, MAPUNT, NOFILE,
•      DRUMAD, DHMWS, PAGESZ, DATFIL, STAFIL, ASAV, ASAVFL
•      NHSTUN, NHSTFI, SCTRUN, MAPFIL
•      DOTUNT, DOTFIL, NCHPAS, TRNSFL, BMTRFL, HISTFL, PCHUNT,
•      CRDUNT, PRDUNT, RANDIO
COMMON /LABS/ NOCAT, CATNAM(60), NOCL2, CLSNM2(60), NOCAT2, CATNM2(60),
•      SUBRAY(120), PTR(60), CATPTR(250), CATDOT(500),
•      DOTVEC(250), COND, MIX, PROC, MAPKEY, DOTKEY, STATKY,
•      SUNANG, T, NEARST, DIST, NOFEAT, FETVEC(30), OMAPUN, OMAFFI,
•      OSAVTP, OSTAFI, NOSUN, ANGLE(8), SIZE, TOTDT2, FLDINF(6),
•      CLSSYM(62), STADRS, MEANAD, TABADR, MAPADR, SUNCOR(30),
•      ODOTUN, ODOTFI, MANSTA, MANDOT, DSPUNT, DSPFIL, DSPKEY, PRNSTS,
•      PRNDOT, FLDNAM, VERTEX(22), NOVRT, NSUN, ANGLES(8)
•      TOTDT3, FLDAOR, VTXAOR
CSEND
C      DIMENSION FLDSAV(4,1), VERTX(2,1), ARRAY(1), SUBVEC(60), SUBNOS(60)
C      DIMENSION N(60), CATVEC(60), SUBNAM(60), CNDSUB(60), MIXSUB(60)
C      REAL MEANS(NOFET2,1)
C      DIMENSION DOTVC2(250), DOTSUM(60,60)
C      DIMENSION TABLE(3135)
C      DATA FLANK/' '/
C      SWITCH = 2
C      INITIALIZE DOTSUM
C      DO 10 I=1,60
C      DO 10 J=1,60
C      DOTSUM(I,J)=0
C      CONTINUE
C      MANNUALLY RELABEL STATS
C      IF (MANSTA .EQ. 0) GO TO 20
C      CALL MANORD (ARRAY (CLSID2), CLSVC2, SUBVEC, NOCLS2, SUBNOS, NOSUB2)
C      UPDATE INFO IN ARRAY
C      CALL REORDER (ARRAY, SUBVEC, N)
C      OUTPUT REORDERED STATS
C      CALL LABMAN (OSAVTP, OSTAFI, NOCLS2, NOSUR2, NOFET2, NOFLD2, TOTVT2,
•      FETVC2, ARRAY (FLDSV2), ARRAY (VERTX2), ARRAY (CLSID2),
•      SUBNOS, ARRAY (SUBDS2), N, STADRS, VARSZ2, PUNCH, SUBVEC, PRNSTS, SWITCH)
C      20 IF (MANDOT .EQ. 0) GO TO 40
C      UPDATE DOTFIL
C      CALL LABDOT (ARRAY (COVAR2))
C      OUTPUT UPDATE DOTFIL
C      CALL WRTDOT (TOTVT2, NOSUN, FLDSAV, VERTX, ANGLE, ARRAY (COVAR2),
•      NOCAT, CATNAM, SIZE, NOFEAT, FETVEC, TOTVRT, NOFLD,
•      ODOTUN, ODOTFI)
C      IF (PRNDOT .EQ. 1) GOTO 798
C      30 CONTINUE
C      EXECUTING A PROEDURE

```

LAB00010
 LAB00020
 LAB00030
 LAB00040
 LAB00050
 LAB00060
 LAB00070
 LAB00080
 LAB00090
 LAB00100
 LAB00110
 LAB00120
 LAB00130
 LAB00140
 LAB00150
 LAB00160
 LAB00170
 LAB00180
 LAB00190
 LAB00200
 LAB00210
 LAB00220
 LAB00230
 LAB00240
 LAB00250
 LAB00260
 LAB00270
 LAB00280
 LAB00290
 LAB00300
 LAB00310
 LAB00320
 LAB00330
 LAB00340
 LAB00350
 LAB00360
 LAB00370
 LAB00380
 LAB00390
 LAB00400
 LAB00410
 LAB00420
 LAB00430
 LAB00440
 LAB00450
 LAB00460
 LAB00470
 LAB00480
 LAB00490
 LAB00500
 LAB00510
 LAB00520
 LAB00530
 LAB00540
 LAB00550
 LAB00560
 LAB00570
 LAB00580
 LAB00590
 LAB00600
 LAB00610
 LAB00620
 LAB00630
 LAB00640
 LAB00650
 LAB00660
 LAB00670
 LAB00680
 LAB00690
 LAB00700
 LAB00710
 LAB00720
 LAB00730
 LAB00740
 LAB00750
 LAB00760

THE

```

C
C
40 IF (PROC.EQ. 3) RETURN
C
C HAVE LABELS BEEN DEFINED
C
C IF (NOCAT.EQ.0) WRITE(6,42)
42 FORMAT( // 25X, '*****CATEGORIES HAVE NOT BEEN DEFINED*****//')
C
C SET UP DOTVEC ARRAY
C
C IF (TOTOT3.EQ. 0) GO TO 49
C K = 0
C TOTDOT = TOTOT2 + TOTOT3
C DO 47 I=1,TOTDOT
C DO 45 J=1,TOTOT3
C IF (DOTVEC(J).EQ. 1) GO TO 47
45 CONTINUE
C K = K + 1
C DOTVC2 (K) = 1
47 CONTINUE
C
C DO 48 I=1,TOTOT2
48 DOTVEC(I) = DOTVC2(I)
C GO TO 52
C
C DO 51 I=1,TOTOT2
51 DOTVEC(I) = 1
52 CONTINUE
C
C STORE FIELD INFO AND VERTICES ON DRUM
C
C IF (DSPKEY.EQ. 0) GO TO 56
C TOTWDS = 4*NOFLD
53 CALL RWRITE(FLDADR,FLDSAV,TOTWDS,ISTAT)
C IF (ISTAT.EQ. 1) GO TO 53
C TOTVTS = TOTVRT*2
C CALL RWRITE(VTXADR,VERTX,TOTVTS,ISTAT)
54 IF (ISTAT.EQ. 1) GO TO 54
56 CONTINUE
C
C
C READ MEANS INTO CORE -- USE SPACE FOR FIELD INFO
C
C MEAN1 = STADRS + VARSZ2*NOSUB2
C TOTWRD = NOFEAT*NOSUB2
C
C CALL RHEAD(MEAN1,MEANS,TOTWRD,ISTAT)
50 IF (ISTAT.EQ. 1) GO TO 50
C
C ZERO OUT SUHNS (WILL CONTAIN NO. OF CLUSTERS IN CATEGORY I)
C
C DO 55 I=1,NOCAT
55 SUHNS(I) = 0
C COMPUTE DISTANCE TABLE
C
C CALL DOTDST(MEANS,ARRAY(COVAR2),TABLE,LIMIT)
C
C K-NEAREST NEIGHBOR PROCEDURE
C
C IF (PROC.NE. 1) GO TO 60
C ITER = NOSUH2
C TAR1 = TABADR
C SWCH = 0
C CALL KNEAR(ARRAY(COVAR2),SUBVEC,SUBNOS,CATVEC,ITER,TAR1,SWCH,
C • DUMMY,DUMMY,MEANS,DOTSUM)
C SWCH = 2
C GO TO 70
C
C ALL-OF-A-KIND
C
60 CALL ALLKIN(ARRAY(COVAR2),SUBVEC,SUBNOS,CATVEC,MEANS,DOTSUM)
C
C ASSIGN APPROPRIATE NAMES TO CLUSTERS

```

~~18-25~~
35%

FILE LABLR

```

C
70 CATNO=NOCAT
   CALL NAMSTA(SURNAM,CATVEC,SUBNOS,NOSUR2,CATNAM,NOCAT)
   IF (CATNO.EQ.NOCAT) GO TO 75
   IF (EXITT.EQ.0) GO TO 75
C
   USER WISHES TO EXIT IF ANY CLASSES NOT USED BY LABEL
C
1000 WRITE(6,1000) NOCAT,CATNO
      FORMAT(//,1X,I3,' LABELS REMAINING OF ',I3,'. EXIT TAKEN')
      CALL EXIT
C
   OUTPUT LABELED STATS
75  * CALL LABMAN(OSAVTP,OSTAFI,NOCAT,NOSUB2,NOFET2,NOFLD2,TOTVT2,
      * FETVC2,ARRAY(FLOSV2),ARRAY(VERTEX2),CATNAM,SUBNOS,SUBNAM,
      * KEPPTS,STADRS,VARSZ2,PUNCH,SUBVEC,PRNSTS,SWTCH)
      IF (COND.EQ.0) GO TO 90
C
   FLAG CONDITIONAL CLUSTERS
   CALL CNDMAP(ARRAY(COVAR2),CNDSUB,CATVEC)
   OUTPUT CONDITIONAL MAP
C
   CALL CLSMAP(CNDSUB,1,SURNOS,SUBVEC,SUBNAM,CATVEC)
90  IF (MIX.EQ.0) GO TO 100
   IF (COND.NE.0) OMAPP1 = OMAPP1 + 1
C
   FLAG MIXED CLUSTERS
   CALL MIXMAP(ARRAY(COVAR2),MIXSUB,NOSUB2,CATVEC)
   OUTPUT MIXED MAP
   CALL CLSMAP(MIXSUB,2,SURNOS,SURVEC,SUBNAM,CATVEC)
   OUTPUT DISPLAY INTERFACE TAPE--MAPTAP
C
100  IF (DSPKEY.EQ.0) GO TO 110
   CALL RHEAD(FLODAD,FLOSAV,TOTWDS,ISTAT)
103  IF (ISTAT.EQ.1) GO TO 103
   CALL RHEAD(VRTXADR,VRTX,TOTVTS,ISTAT)
105  IF (ISTAT.EQ.1) GO TO 105
C
   CALL DSPTAP(SURNOS,SUBNAM,FLOSAV,VRTX,CATVEC,SUBVEC,MEANS,
   * ARRAY(COVAR2),TOP,ARRAY,NOFLD,TOTVRT)
110 CONTINUE
   RETURN
C
   CODE TO PRINT DOT DATA RECORD
C
798 CONTINUE
   ISTART=1
   IEND=10
799 CONTINUE
   IKT=0
   DO 800 I=1,TOTD/2
      IKT=IKT+1
      IF (IEND.GT.NOFEAT) IEND=NOFEAT
      IF (I.NE.1.AND.IKT.EQ.1) WRITE(6,810)
810  FORMAT(1H,5(/))
      IF (IKT.NE.1) GO TO 820
      WRITE(6,700)
700  FORMAT(//)
      WRITE(6,690)
690  FORMAT(1X,'NO.',2X,'SAMPLE',2X,'LINE',2X,'TYPE',2X,'CATEGORY',
      * 30X,'(DATA)')
      WRITE(6,720) (HLANK,FETVC2(I),I=ISTART,IEND)
720  FORMAT(37X,10(A1,'CH',I2,''))
820 CONTINUE

```

LAB01530
 LAB01540
 LAB01550
 LAB01560
 LAB01570
 LAB01580
 LAB01590
 LAB01600
 LAB01610
 LAB01620
 LAB01630
 LAB01640
 LAB01650
 LAB01660
 LAB01670
 LAB01680
 LAB01690
 LAB01700
 LAB01710
 LAB01720
 LAB01730
 LAB01740
 LAB01750
 LAB01760
 LAB01770
 LAB01780
 LAB01790
 LAB01800
 LAB01810
 LAB01820
 LAB01830
 LAB01840
 LAB01850
 LAB01860
 LAB01870
 LAB01880
 LAB01890
 LAB01900
 LAB01910
 LAB01920
 LAB01930
 LAB01940
 LAB01950
 LAB01960
 LAB01970
 LAB01980
 LAB01990
 LAB02000
 LAB02010
 LAB02020
 LAB02030
 LAB02040
 LAB02050
 LAB02060
 LAB02070
 LAB02080
 LAB02090
 LAB02100
 LAB02110
 LAB02120
 LAB02130
 LAB02140
 LAB02150
 LAB02160
 LAB02170
 LAB02180
 LAB02190
 LAB02200
 LAB02210
 LAB02220
 LAB02230
 LAB02240
 LAB02250
 LAB02260
 LAB02270
 LAB02280

FILE LABLR

| | | |
|-----|---|----------|
| | IDUM=COVAR2*(II-1)*SIZE-1 | LAB02290 |
| | IDMM=COVAR2*(II-1)*SIZE+3 | LAB02300 |
| | WRITE(6,710)II,(ARRAY(IDUM+1),I=1,4),(ARRAY(IDMM+JJ),JJ=ISTART,IEND | LAB02310 |
| | *D) | LAB02320 |
| 710 | FORMAT(1X,I3,1H,3X,I4,3X,I4,2X,I2,6X,I2,8X,10(I3,4X)) | LAB02330 |
| | WRITE(6,712) | LAB02340 |
| 712 | FORMAT() | LAB02350 |
| 800 | CONTINUE | LAB02360 |
| | IF(NOFEAT.GT.10) GO TO 830 | LAB02370 |
| | GO TO 840 | LAB02380 |
| 830 | CONTINUE | LAB02390 |
| | IF(ITWO.EQ.1) GO TO 840 | LAB02400 |
| | ITWO=1 | LAB02410 |
| | ISTART=IEND+1 | LAB02420 |
| | IEND=NOFEAT | LAB02430 |
| | GO TO 799 | LAB02440 |
| 840 | CONTINUE | LAB02450 |
| | GOTO 30 | LAB02460 |
| | END | LAB02470 |

ORIGINAL PAGE 1
OF POOR QUALITY

FILE: MANORD

```

C      SUBROUTINE MANORD (CLSNAM, CLSVEC, SUBVEC, NOCLS2, SUBNO, NOSUB2)
C      MANORD REGROUPS THE SUBCLASS IN THE ARRAY SUBVEC
C      IMPLICIT INTEGER (A-Z)
C      INCLUDE CMRK15.LIST
COMMON /LAHS/ NOCAT, CATNAM(60), NOCL2, CLSNM2(60), NOCAT2, CATNM2(60),
*      SURREY(120), PTR(60), CATPTR(250), CATDOT(500),
*      DOTVEC(250), COMD, MIX, PROC, MAPKEY, DOTKEY, STATKY,
*      SUNANG, T, NEARST, DIST, NOFEAT, FEIVC(30), OMAPUN, OMAPI,
*      OSAVTP, OSTAFI, NOSUN, ANGLE(8), SIZE, TOTDT2, FLDINF(6),
*      CLSSYM(62), STADRS, MEANAD, TABADR, MAPADR, SUNCOR(30),
*      ODOTUN, ODOTFI, MANSTA, MANDOT, DSPUNT, DSPFIL, DSPKEY, PRNSTS,
*      PRNDOT, FLONAM, VERTEX(22), NOVRT, NSUN, ANGLE5(8)
*      , TOTDT3, FLOADR, VTXADR
C$END
DIMENSION SURNO(60), CLSVEC(1), CLSNAM(1), SUBVEC(1)
DIMENSION CLSNO(60)

CHECK VALIDITY OF CLASS NAMES
DO 150 I=1, NOCL2
DO 120 J=1, NOCLS2
IF (CLSNM2(I) .EQ. CLSNAM(J)) GO TO 145
120 CONTINUE
WRITE(6,130) (CLSNAM(K), K=1, NOCLS2)
130 FORMAT(1, 'ERROR IN INPUT OF CLASS NAMES. NAMES ON STAT FILE ARE :',
*      (7(A4.3X) / T55))
WRITE(6,140) (CLSNM2(K), K=1, NOCL2)
140 FORMAT(1, 'NAMES INPUT ARE :', (10(A4.3X) /))
145 CLSNO(I) = J
150 CONTINUE

REARRANGE SUBVC2 ARRAY SO ALL SUBCLASSES FOR A GIVEN CLASS ARE
GROUPED TOGETHER
DO 160 I=1, NOCL2
REG. AND END PTRS FOR SURREY
IB = PTR(I) + 1
IE = IB + SUBRAY(IB-1) - 1
DO 160 J=IB, IE
STORE NEW CLASS NO FOR RELABELED SUBCLASS
M = SURREY(J)
160 CLSVEC(M) = CLSNO(I)
ZERO SUBNO ARRAY
DO 165 I=1, NOSUB2
165 SURNO(I) = 0
COMPUTE NO. OF SUBCLASSES IN EACH NEW CLASS
DO 170 I=1, NOSUB2
K = CLSVEC(I)
170 SURNO(K) = SURNO(K) + 1
ORDER THE SUBCLASS NOS. ACCORDING TO THE
NEWLY ASSIGNED CLASS NO -- STORE IN SURVEC.
KK = 0
DO 180 I=1, NOCLS2
DO 180 J=1, NOSUB2
IF (CLSVEC(J) .NE. I) GO TO 180
KK = KK + 1
SURVEC(KK) = J
180 CONTINUE
RETURN

```

MAN00010
 MAN00020
 MAN00030
 MAN00040
 MAN00050
 MAN00060
 MAN00070
 MAN00080
 MAN00090
 MAN00100
 MAN00110
 MAN00120
 MAN00130
 MAN00140
 MAN00150
 MAN00160
 MAN00170
 MAN00180
 MAN00190
 MAN00200
 MAN00210
 MAN00220
 MAN00230
 MAN00240
 MAN00250
 MAN00260
 MAN00270
 MAN00280
 MAN00290
 MAN00300
 MAN00310
 MAN00320
 MAN00330
 MAN00340
 MAN00350
 MAN00360
 MAN00370
 MAN00380
 MAN00390
 MAN00400
 MAN00410
 MAN00420
 MAN00430
 MAN00440
 MAN00450
 MAN00460
 MAN00470
 MAN00480
 MAN00490
 MAN00500
 MAN00510
 MAN00520
 MAN00530
 MAN00540
 MAN00550
 MAN00560
 MAN00570
 MAN00580
 MAN00590
 MAN00600
 MAN00610
 MAN00620
 MAN00630
 MAN00640
 MAN00650
 MAN00660
 MAN00670
 MAN00680
 MAN00690
 MAN00700
 MAN00710
 MAN00720
 MAN00730
 MAN00740
 MAN00750
 MAN00760
 MAN00770
 MAN00780
 MAN00790

FILE: MANORD

C

END

MAN00800
MAN00810

ORIGINAL PAGE IS
OF POOR QUALITY

~~18-29~~

360

FILE: MAPHND

```

C      THIS ROUTINE PRINTS THE HEADER INFORMATION FOR THE CLASSIFICATION MAP IN CLASSIFY AND DISPLAY
C
C      SURROUTINE MAPHND(INOCAT,CLSSYM,CATNAM,KATNO,SURDES,CATSUB)
C          NOCAT -- NO. OF CATEGORIES
C          CLSSYM -- SYMBOLS FOR CATEGORIES OR SURCLASSES
C          CATNAM -- CATEGORY NAMES
C          KATNO -- CATEGORY EACH CLASS WAS ASSIGNED TO
C          CLSMTX -- CLASS NAMES
C          SURNO -- NO. OF SURCLASSES IN EACH CLASS
C          SURDES -- SURCLASS NAMES
C          CLSVC2 -- CLASS EACH SUBCLASS WAS ASSIGNED TO (IN COMMON BLOCK INFORM)
C
C      IMPLICIT INTEGER (A-Z)
C
C      INCLUDE COMR1,LIST
C      COMMON/INFORM/NOCLS2,NOSUR2,NOFET2,VARSZ2,TOTVT2,NOFLD2,
C      *      AVAR2,COVAR2,CLSID2,SURN02,SURDS2,FLDSV2,VERTX2,
C      *      FETVC2(30),SURVC2(75),SURPTR(75),CLSVC2(60),
C      *      KEPPTS(60),NOGRP,GRPNAM(60),GRPDEX(61),
C      *      GRPCHK(61),GROUPS(124)
CSEND
C
C      LOGICAL ISWTH
C      DIMENSION CLSSYM(1),CATNAM(1),KATNO(1),SUBDES(1)
C      DIMENSION CATSUB(1)
C
C      PRINTS CATEGORY CLASSIFIER INFORMATION
C
C      WRITE(6,200)
200  FORMAT(/T54,'LBELED CLUSTER MAP',/T49,'( ** - DENOTES MIXED/COND
* CLUSTER )',/T50,' * - DENOTES DU/DU AREA', 9X,')',/T33,'LBELED',
* T41,'SURCLASS',/T31,'NO.',T35,'NAME',T64,'UNLBELED NO.',T40,
* 'LBELED NO.',T94,'NAME',T101,'SYMBOL')
C
C      JJ = 0
C      KK = 0
C      DO 68 I=1,NOCAT
C
C      WRITE(6,210)I,CATNAM(I)
210  FORMAT(/T31,I2,T37,A4)
C      ISWTH = .TRUE.
C      DO 63 J=1,NOSUR2
C      IF (KATNO(J) .EQ. I) GO TO 64
C      GO TO 63
C      64  KK = KK + 1
C      IF (ISWTH) GO TO 72
C      WRITE(6,250)J,KK,SURDES(KK),CLSSYM(J)
250  FORMAT(  T64,I2,T84,I2,T94,A4,T105,A1)
C      GO TO 75
C      72  WRITE(6,240)J,KK,SURDES(KK),CLSSYM(J)
240  FORMAT(1H+,T69,I2,T84,I2,T94,A4,T105,A1)
C      ISWTH = .FALSE.
C      75  IF (CATSUB(J) .GT. (NOCAT + 2)) WRITE(6,80)
C      80  FORMAT(1H+,T92,2H**)
C      63  CONTINUE
C      68  CONTINUE
C      RETURN
C      END

```

MAP00010
 MAP00020
 MAP00030
 MAP00040
 MAP00050
 MAP00060
 MAP00070
 MAP00080
 MAP00090
 MAP00100
 MAP00110
 MAP00120
 MAP00130
 MAP00140
 MAP00150
 MAP00160
 MAP00170
 MAP00180
 COM00010
 COM00020
 COM00030
 COM00040
 COM00050
 MAP00200
 MAP00210
 MAP00220
 MAP00230
 MAP00240
 MAP00250
 MAP00260
 MAP00270
 MAP00280
 MAP00290
 MAP00300
 MAP00310
 MAP00320
 MAP00330
 MAP00340
 MAP00350
 MAP00360
 MAP00370
 MAP00380
 MAP00390
 MAP00400
 MAP00410
 MAP00420
 MAP00430
 MAP00440
 MAP00450
 MAP00460
 MAP00470
 MAP00480
 MAP00490
 MAP00500
 MAP00510
 MAP00520
 MAP00530
 MAP00540
 MAP00550
 MAP00560
 MAP00570

[illegible]

ORIGINAL PAGE IS
OF POOR QUALITY

FILE: REORDER

| | | |
|--------|---|----------|
| C | SURROUTINE REORDER (ARRAY, SURVEC, N) | RE000010 |
| C | REORDER ID INFORMATION IN ARRAY | RE000020 |
| C | IMPLICIT INTEGER (A-Z) | RE000030 |
| C | INCLUDE COMMON LIST | RE000040 |
| C | COMMON/INFORM/NOCLS2, NOSUR2, NOFET2, VARS22, TOTVT2, NOFLD2, | RE000050 |
| | AV4R2, COVAR2, CLSID2, SURNO2, SURDS2, FLDSV2, VERTX2, | RE000060 |
| | FETVC2(30), SUHVC2(75), SUHPTR(75), CLSVC2(60), | COM00010 |
| | KEPPTS(60), NOGRP, GRPNAM(60), GRPDEX(61), | COM00020 |
| | GRPCHK(61), GROUPS(124) | COM00030 |
| C\$END | DIMENSION ARRAY(1), SUBNAM(60), N(1) | COM00040 |
| | DIMENSION SURVEC(1) | COM00050 |
| C | UPDATE SURCLASS NAMES | RE000080 |
| C | DO 100 I=1, NOSUR2 | RE000090 |
| | K = SURVEC(I) | RE000100 |
| 100 | SUBNAM(I) = ARRAY(SUBDS2 + K-1) | RE000110 |
| | DO 110 I=1, NOSUR2 | RE000120 |
| 110 | ARRAY(SUBDS2 + I-1) = SUBNAM(I) | RE000130 |
| C | UPDATE POPULATION ARRAY | RE000140 |
| | DO 130 I=1, NOSUR2 | RE000150 |
| | K = SURVEC(I) | RE000160 |
| 130 | N(I) = KEPPTS(K) | RE000170 |
| | RETURN | RE000180 |
| C | END | RE000190 |
| | | RE000200 |
| | | RE000210 |
| | | RE000220 |
| | | RE000230 |
| | | RE000240 |
| | | RE000250 |
| | | RE000260 |
| | | RE000270 |
| | | RE000280 |
| | | RE000290 |

FILE SET14

```

SUBROUTINE SET14 (ARRAY, TOP, EXIT)
C*
C* SET14 READS IN THE CONTROL CARDS FOR THE LABEL PROCESSOR
C*
IMPLICIT INTEGER (A-Z)
DIMENSION CODE (21), CARD (62), EQUOM (3)
DATA CODE / 'CHAN', 'DATA', 'MAPF', 'DOTF', 'OPTI', 'EXCL',
* 'STAT', 'DOTL', 'STAL', 'DIST', 'THRE', 'NEAR', 'PROC', 'MODU',
* 'SUNA', 'MAPT', 'DATE', 'COMM', 'HED1', 'HED2', 'END' /
C IN THE CONVERSION FROM 1108 TO IBM, STATLA BECOMES STAL
DIMENSION ARRAY (1), ACARD (20)
DIMENSION SLASH (2), EQUVEC (2)
DATA SLASH / 1, '/' /
DATA EQUVEC / 1, '=' /
DATA EQUOM / 2, '=' /
DATA CODE1 / 1, '/' /
C*
C* INCLUDE COMRK1, LIST
C* INCLUDE COMRK4, LIST
C* INCLUDE COMRK6, LIST
C* INCLUDE COMRK15, LIST
COMMON / INFORM / NOCL2, NOSUB2, NOFET2, VARSZ2, TOTVT2, NOFLD2,
* AVAR2, COVAR2, CLS102, SUBNO2, SUBDS2, FLDSV2, VERTX2,
* FETVC2 (30), SURVC2 (75), SURPTR (75), CLSVC2 (60),
* KEPTS (60), NOGRP, GRPNAM (60), GRPDEX (61),
* GRPCHK (61), GROUPS (124)
DIMENSION HED1 (15), HED2 (15), DATE (3), COMMENT (15)
EQUIVALENCE (HED1 (1), HEAD (4)), (DATE (1), HEAD (22)),
2 (HED2 (1), HEAD (30)), (COMMENT (1), HEAD (48))
COMMON / GLOHAL / HEAD (63), MAPTAP, DATEF, SAVTAP, RMFILE, RMKEY,
* HISFIL, HISKEY, TRFORM, FRIPTR, ERKEY, MAPUNT, NOFILE,
* DRUMAD, DRUMDS, PAGESZ, DATFIL, STAFIL, ASAV, ASAVFL
* NHSTUN, NHSTFI, SCTRUN, MAPFIL
* DOTUNT, DOTFIL, NCHPAS, TRNSFL, RMTRFL, HISTFL, PCHUNT,
* CHDUNT, PRUNT, RANDIO
COMMON / LAHS / NOCAT, CATNAM (60), NOCL2, CLSNM2 (60), NOCAT2, CATNM2 (60),
* SURRAY (120), PTR (60), CATPTR (250), CATDOT (500),
* DOTVEC (250), COND, MIX, PROC, MAPKEY, DOTKEY, STATKY,
* SUNANG, T, NEARST, DIST, NOFEAT, FETVEC (30), OMAPUN, OMAFFI,
* OSAVTP, OSTAFI, NOSUN, ANGLE (8), SIZE, TOTDT2, FLDINF (6),
* CLSSYM (62), STADRS, MEANAD, TABADR, MAPADR, SUNCOR (30),
* ODOTUN, ODOTFI, MANSTA, MANDOT, DSPUNT, DSPFIL, DSPKEY, PRNSTS,
* PRNDOT, FLDNAM, VERTEX (22), NOVPT, NSUN, ANGLES (8)
* TOTDT3, FLDADR, VTADR
C$END
REAL T, SUNCOR
DATA BLANK / ' ', SBCD / 'S', DBCD / 'D', UBCD / 'U',
* FACD / 'F', IRCU / 'I', ORCD / 'O', COMMA / ',', CBCD / 'C',
* MACD / 'M', BCD1 / '1', BCD2 / '2', KCD / 'K', ABCD / 'A' /
C DATA EBCD / 'E' /
C INITIALIZE PARAMETERS
NPUT = 21
TOTDT3 = 0
COND = 0
MIX = 0
PROC = 1
PRNDOT = 0
PRNSTS = 0
MANSTA = 0
MANDOT = 0
MAPKEY = 0
DOTKEY = 0
DSPKEY = 0
STATKY = 0
NOSUN = 0
SUNANG = 0
T = 25.0
NEARST = 1
DIST = 1
NOFET2 = 0
NOFEAT = 0
PTR1 = 0
PTR2 = 0
NOCAT2 = 0
NOCL2 = 0
EXIT = 0

```

ORIGINAL PAGE IS
OF POOR QUALITY

FILE SET14

```

C
C
C   CLSVC2(1) = BLANK
C   SUBPTR(75) = BLANK
C
C   DO 20 I=1,30
C   20 SUNCOR(I) = 1.0
C   WRITE(6,100)
C   100 FORMAT(/11X,'INPUT SUMMARY'//)
C
C   PUT THE NEXT CARD IN THE REREAD BUFFER
C
C   RRUNIT=30
C   105 READ(21,107)(ACARD(I),I=1,20)
C   107 FORMAT(20A4)
C   WRITE(30,107)(ACARD(I),I=1,20)
C   REWIND RRUNIT
C   READ(30,110)CODE1,CARD
C   REWIND RRUNIT
C   COL = 0
C   WRITE(6,120)CODE1,CARD
C   110 FORMAT (A4, 6X, 62A1)
C   120 FORMAT (1X, A4, 6X, 62A1)
C   DO 130 I=1,NPUT
C   * IF (CODE1.EQ.CODE(I)) GO TO (150,180,210,250,330,365,370,
C   * 390,400,410,420,430,440,460,470,490,510,530,540,550,560).I
C   130 CONTINUE
C   135 WRITE(6,140)
C   140 FORMAT(' INVALID CONTROL CARD - IGNORED ')
C   GO TO 105
C
C   CHANNEL CARD
C
C   150 M=NXTCHR(CARD,COL)
C   IF (M.EQ. BLANK) GO TO 105
C   IF (M.EQ. SBCD) GO TO 160
C   IF (M.EQ. DBCD) GO TO 170
C   153 WRITE(6,155)
C   155 FORMAT(' ERROR ON CHANNELS CARD')
C   GO TO 105
C   160 J = FIND12(CARD,COL,EQUCOM)
C   IF (J.NE. 2) GO TO 153
C   NOFET2 = NUMBER(CARD,COL,FETVC2,NOFET2)
C   COL = COL - 1
C   CALL ORDER(FETVC2,NOFET2)
C   GO TO 150
C   170 J = FIND12(CARD,COL,EQUCOM)
C   IF (J.NE. 2) GO TO 153
C   NOFEAT = NUMBER(CARD,COL,FETVEC,NOFEAT)
C   COL = COL - 1
C   CALL ORDER(FETVEC,NOFEAT)
C   GO TO 150
C
C   DATA FILE CARD
C
C   180 M = NXTCHR(CARD,COL)
C   IF (M.EQ. BLANK) GO TO 105
C   IF (M.EQ. UHCD) GO TO 190
C   IF (M.EQ. FRCD) GO TO 200
C   185 WRITE(6,187)
C   187 FORMAT(' ERROR ON DATA FILE CARD')
C   GO TO 105
C   190 J = FIND12(CARD,COL,EQUCOM)
C   IF (J.NE. 2) GO TO 185
C   M = NUMBER(CARD,COL,DATAPE,ZERO)
C   COL = COL - 1
C   GO TO 180
C   200 J = FIND12(CARD,COL,EQUCOM)
C   IF (J.NE. 2) GO TO 185
C   M = NUMBER(CARD,COL,DATFIL,ZERO)
C   DATFIL = DATFIL - 1
C   COL = COL - 1
C   GO TO 180
C
C   CLUSTER/CLASSIFICATION MAP TAPE

```

SET00770
 SET00780
 SET00790
 SET00800
 SET00810
 SET00820
 SET00830
 SET00840
 SET00850
 SET00860
 SET00870
 SET00880
 SET00890
 SET00900
 SET00910
 SET00920
 SET00930
 SET00940
 SET00950
 SET00960
 SET00970
 SET00980
 SET00990
 SET01000
 SET01010
 SET01020
 SET01030
 SET01040
 SET01050
 SET01060
 SET01070
 SET01080
 SET01090
 SET01100
 SET01110
 SET01120
 SET01130
 SET01140
 SET01150
 SET01160
 SET01170
 SET01180
 SET01190
 SET01200
 SET01210
 SET01220
 SET01230
 SET01240
 SET01250
 SET01260
 SET01270
 SET01280
 SET01290
 SET01300
 SET01310
 SET01320
 SET01330
 SET01340
 SET01350
 SET01360
 SET01370
 SET01380
 SET01390
 SET01400
 SET01410
 SET01420
 SET01430
 SET01440
 SET01450
 SET01460
 SET01470
 SET01480
 SET01490
 SET01500
 SET01510
 SET01520

FILE SET14

```

210 M = NXTCHR(CARD,COL)
211 IF (M.EQ. IHCD) GO TO 213
    IF (M.EQ. OHCD) GO TO 220
    IF (M.EQ. BLANK) GO TO 105
212 WRITE(6,216)
216 FORMAT(' ERROR ON MAPFIL CARD')
    GO TO 105
C
213 J = FIND12(CARD,COL,SLASH)
    IF (J.EQ. -1) GO TO 212
    MAPKEY = 1
214 M = NXTCHR(CARD,COL)
    IF (M.EQ. COMMA) GO TO 214
    IF (M.EQ. FRCD) GO TO 215
    IF (M.EQ. URCD) GO TO 211
    J = FIND12(CARD,COL,EQUVEC)
    IF (J.EQ. -1) GO TO 212
    M = NUMBER(CARD,COL,MAPUNT,ZERO)
    COL = COL - 1
    GO TO 214
215 J = FIND12(CARD,COL,EQUVEC)
    IF (J.EQ. -1) GO TO 212
    M = NUMBER(CARD,COL,MAPFIL,ZERO)
    COL = COL - 1
    MAPFIL = MAPFIL - 1
    GO TO 214
220 J = FIND12(CARD,COL,SLASH)
    IF (J.EQ. -1) GO TO 212
221 M = NXTCHR(CARD,COL)
    IF (M.EQ. COMMA) GO TO 221
    IF (M.EQ. FRCD) GO TO 222
    IF (M.EQ. URCD) GO TO 211
    J = FIND12(CARD,COL,EQUVEC)
    IF (J.EQ. -1) GO TO 212
    M = NUMBER(CARD,COL,OMAPUN,ZERO)
    COL = COL - 1
    GO TO 221
222 J = FIND12(CARD,COL,EQUVEC)
    IF (J.EQ. -1) GO TO 212
    M = NUMBER(CARD,COL,OMAPFI,ZERO)
    COL = COL - 1
    OMAPI = OMAPI - 1
    GO TO 221
C
DOTFIL CARD
C
250 M = NXTCHR(CARD,COL)
251 IF (M.EQ. IHCD) GO TO 254
    IF (M.EQ. OHCD) GO TO 260
    IF (M.EQ. BLANK) GO TO 105
252 WRITE(6,253)
253 FORMAT(' ERROR ON MAPFILE CARD')
    GO TO 105
C
254 J = FIND12(CARD,COL,SLASH)
    IF (J.EQ. -1) GO TO 252
    DOTKEY = 1
256 M = NXTCHR(CARD,COL)
    IF (M.EQ. COMMA) GO TO 256
    IF (M.EQ. FRCD) GO TO 258
    IF (M.EQ. URCD) GO TO 251
    J = FIND12(CARD,COL,EQUVEC)
    IF (J.EQ. -1) GO TO 252
    M = NUMBER(CARD,COL,DOTUNT,ZERO)
    COL = COL - 1
    GO TO 256
258 J = FIND12(CARD,COL,EQUVEC)
    IF (J.EQ. -1) GO TO 252
    M = NUMBER(CARD,COL,DOTFIL,ZERO)
    COL = COL - 1
    DOTFIL = DOTFIL - 1
    GO TO 256
C
260 J = FIND12(CARD,COL,SLASH)
    IF (J.EQ. -1) GO TO 252
262 M = NXTCHR(CARD,COL)

```

SET01530
 SET01540
 SET01550
 SET01560
 SET01570
 SET01580
 SET01590
 SET01600
 SET01610
 SET01620
 SET01630
 SET01640
 SET01650
 SET01660
 SET01670
 SET01680
 SET01690
 SET01700
 SET01710
 SET01720
 SET01730
 SET01740
 SET01750
 SET01760
 SET01770
 SET01780
 SET01790
 SET01800
 SET01810
 SET01820
 SET01830
 SET01840
 SET01850
 SET01860
 SET01870
 SET01880
 SET01890
 SET01900
 SET01910
 SET01920
 SET01930
 SET01940
 SET01950
 SET01960
 SET01970
 SET01980
 SET01990
 SET02000
 SET02010
 SET02020
 SET02030
 SET02040
 SET02050
 SET02060
 SET02070
 SET02080
 SET02090
 SET02100
 SET02110
 SET02120
 SET02130
 SET02140
 SET02150
 SET02160
 SET02170
 SET02180
 SET02190
 SET02200
 SET02210
 SET02220
 SET02230
 SET02240
 SET02250
 SET02260
 SET02270
 SET02280

ORIGINAL PAGE IS
OF POOR QUALITY

FILE SET14

```

      IF (M.EQ.COMMA)GO TO 262
      IF (M.EQ.FRCD) GO TO 264
      IF (M.NE.UHCD) GO TO 251
      J = FIND12(CARD,COL,EQUVEC)
      IF (J.EQ.-1) GO TO 252
      M = NUMBER(CARD,COL,ODOTUN,ZERO)
      COL = COL - 1
      GO TO 262
264 J = FIND12(CARD,COL,EQUVEC)
      IF (J.EQ.-1) GO TO 252
      M = NUMBER(CARD,COL,ODOTFI,ZERO)
      COL = COL - 1
      ODOTFI = ODOTFI - 1
      GO TO 262

```

OPTION CARD

```

C
C
C
330 M = NXTCHR(CARD,COL)
      IF (M.EQ.BLANK)GO TO 105
      IF (M.EQ.CBCD) GO TO 340
      IF (M.EQ.SRCD) GO TO 345
      IF (M.EQ.DRCD) GO TO 350
      IF (M.EQ.MBCD) GO TO 363
      IF (M.EQ.EHCD)GO TO 355
333 WRITE(6,335)
335 FORMAT(' ERROR ON OPTION CARD')
      GO TO 105
340 COND = 1
      GO TO 360
345 PRNSTS = 1
      GO TO 360
350 PRNDOT = 1
      GO TO 360
355 EXIT=1
      GO TO 360
363 MIX = 1
360 J = FIND12(CARD,COL,EQUCOM)
      IF (J.EQ.3) GO TO 330
      IF (J.EQ.-1) GO TO 105

```

EXCLUDE CARD

```

C
C
C
365 TOTDT3 = NUMBER(CARD,COL,DOTVEC,TOTDT3)
      CALL ORDER(DOTVEC,TOTDT3)
      GO TO 105

```

STATFILE CARD

```

C
C
C
370 M = NXTCHR(CARD,COL)
371 IF (M.EQ.IBCD) GO TO 374
      IF (M.EQ.OCDD) GO TO 380
      IF (M.EQ.BLANK)GO TO 105
372 WRITE(6,373)
373 FORMAT(' ERROR ON STATFI CARD')
      GO TO 105
C
374 J = FIND12(CARD,COL,SLASH)
      IF (J.EQ.-1) GO TO 372
      STATKY = 1
375 M = NXTCHR(CARD,COL)
      IF (M.EQ.COMMA)GO TO 375
      IF (M.EQ.FBCD) GO TO 376
      IF (M.NE.UHCD) GO TO 370
      J = FIND12(CARD,COL,EQUVEC)
      IF (J.EQ.-1) GO TO 372
      M = NUMBER(CARD,COL,SAVTAP,ZERO)
      COL = COL - 1
      GO TO 375
376 J = FIND12(CARD,COL,EQUVEC)
      IF (J.EQ.-1) GO TO 372
      M = NUMBER(CARD,COL,STAFIL,ZERO)
      COL = COL - 1
      STAFIL = STAFIL - 1
      GO TO 375
380 J = FIND12(CARD,COL,SLASH)

```

SET02290
 SET02300
 SET02310
 SET02320
 SET02330
 SET02340
 SET02350
 SET02360
 SET02370
 SET02380
 SET02390
 SET02400
 SET02410
 SET02420
 SET02430
 SET02440
 SET02450
 SET02460
 SET02470
 SET02480
 SET02490
 SET02500
 SET02510
 SET02520
 SET02530
 SET02540
 SET02550
 SET02560
 SET02570
 SET02580
 SET02590
 SET02600
 SET02610
 SET02620
 SET02630
 SET02640
 SET02650
 SET02660
 SET02670
 SET02680
 SET02690
 SET02700
 SET02710
 SET02720
 SET02730
 SET02740
 SET02750
 SET02760
 SET02770
 SET02780
 SET02790
 SET02800
 SET02810
 SET02820
 SET02830
 SET02840
 SET02850
 SET02860
 SET02870
 SET02880
 SET02890
 SET02900
 SET02910
 SET02920
 SET02930
 SET02940
 SET02950
 SET02960
 SET02970
 SET02980
 SET02990
 SET03000
 SET03010
 SET03020
 SET03030
 SET03040

FILE SET14

```

      IF (J.EQ.-1) GO TO 372
381 M = NXTCHR(CARD,COL)
      IF (M.EQ.COMMA) GO TO 381
      IF (M.EQ.FHCD) GO TO 382
      IF (M.NE.UHCD) GO TO 370
      J = FIND12(CARD,COL,EQUVEC)
      IF (J.EQ.-1) GO TO 372
      M = NUMRER(CARD,COL,OSAVTP,ZERO)
      COL = COL - 1
      GO TO 381
382 J = FIND12(CARD,COL,EQUVEC)
      IF (J.EQ.-1) GO TO 372
      M = NUMBER(CARD,COL,OSTAFI,ZERO)
      COL = COL - 1
      OSTAFI = OSTAFI - 1
      GO TO 381

C
C   DOTLABEL CARD
390 M = CROSCN(CARD,CATPTR,CATNM2,CATDOT,NOCAT2,PTR1)
      MANDOT = 1
      GO TO 105

C
C   STATLABEL CARD
400 M = CROSCN(CARD,PTR,CLSNM2,SUBRAY,NOCL2,PTR2)
      MANSTA = 1
      GO TO 105

C
C   DISTANCE
410 M = NXTCHR(CARD,COL)
      M = NXTCHR(CARD,COL)
      IF (M.EQ.HCD1) DIST = 1
      IF (M.EQ.RCD2) DIST = 2
      GO TO 105

C
C   THRESHOLD CARD
420 M = FLTNUM(CARD,COL,T,1)
      GO TO 105

C
C   K NEAREST DOTS
430 J = NUMBER(CARD,COL,NEARST,ZERO)
      GO TO 105

C
C   PROCEDURE CARD
      K = NEAREST PROCEDURE = 1
      ALL-OF-A-KIND = 2
      MANUAL LABELING OF FILES = 3

C
C   440 M = NXTCHR(CARD,COL)
      IF (M.EQ.KRCD) GO TO 446
      IF (M.EQ.AHCD) GO TO 448
      IF (M.EQ.MBCD) GO TO 450
      WRITE(6,445)
445 FORMAT(' ERROR ON PROCEDURE CARD')
      GO TO 105
446 PROC = 1
      GO TO 105
448 PROC = 2
      GO TO 105
450 PROC = 3
      GO TO 105

C
C   MODULE CARD DECK
460 CALL CRUSTA(ARRAY, TOP)
      STATKY = 1
      GO TO 105

C
C   SUN ANGLE CARD

```

SET03050
 SET03060
 SET03070
 SET03080
 SET03090
 SET03100
 SET03110
 SET03120
 SET03130
 SET03140
 SET03150
 SET03160
 SET03170
 SET03180
 SET03190
 SET03200
 SET03210
 SET03220
 SET03230
 SET03240
 SET03250
 SET03260
 SET03270
 SET03280
 SET03290
 SET03300
 SET03310
 SET03320
 SET03330
 SET03340
 SET03350
 SET03360
 SET03370
 SET03380
 SET03390
 SET03400
 SET03410
 SET03420
 SET03430
 SET03440
 SET03450
 SET03460
 SET03470
 SET03480
 SET03490
 SET03500
 SET03510
 SET03520
 SET03530
 SET03540
 SET03550
 SET03560
 SET03570
 SET03580
 SET03590
 SET03600
 SET03610
 SET03620
 SET03630
 SET03640
 SET03650
 SET03660
 SET03670
 SET03680
 SET03690
 SET03700
 SET03710
 SET03720
 SET03730
 SET03740
 SET03750
 SET03760
 SET03770
 SET03780
 SET03790
 SET03800

ORIGINAL IS
 IN THE QUANTITY

FILE SET14

```

C
C      SUNANG = 1 -- ANGLES ARE ON DOTFIL
C      SUNANG = 2 -- ANGLES ARE ON CARDS
470 M = NATCHR(CARD,COL)
    IF (M.EQ. FRCU) GO TO 475
    SUNANG = 1
    GO TO 105
475 COL = COL - 1
    IF (M.NE. FRCU) GO TO 475
    SUNANG = 2
    GO TO 105

C
C      MAPTAP CARD --DISPLAY INTERFACE TAPE
490 M = NATCHR(CARD,COL)
    IF (M.EQ. OBCU) GO TO 496
    IF (M.EQ. BLANK) GO TO 105
492 WRITE(6,494)
494 FORMAT(' ERROR ON MAPTAP CARD')
    GO TO 105
496 J = FIND12(CARD,COL,SLASH)
    IF (J.EQ. -1) GO TO 492
    DSPKEY = 1
498 M = NATCHR(CARD,COL)
    IF (M.EQ. COMMA) GO TO 498
    IF (M.EQ. FRCU) GO TO 500
    IF (M.NE. UHCU) GO TO 491
    J = FIND12(CARD,COL,EQUVEC)
    IF (J.EQ. -1) GO TO 492
    M = NUMBER(CARD,COL,DSPUNT,ZERO)
    COL = COL - 1
    GO TO 498
500 J = FIND12(CARD,COL,EQUVEC)
    IF (J.EQ. -1) GO TO 492
    M = NUMBER(CARD,COL,DSPFIL,ZERO)
    DSPFIL = DSPFIL - 1
    COL = COL - 1
    GO TO 498
    DATE CARD

C
C      510 M = NATCHR(CARD,COL)
C          IF (M.EQ. BLANK) GO TO 105
C          READ(30,520) DATE
C          REWIND RRUNIT
520 FORMAT(10X,15A4)
    GO TO 105

C
C      COMMENT CARD
530 M = NATCHR(CARD,COL)
    IF (M.EQ. BLANK) GO TO 105
    READ(30,520) COMMENT
    REWIND RRUNIT
    GO TO 105

C
C      MED1
540 M = NATCHR(CARD,COL)
    IF (M.EQ. BLANK) GO TO 105
    READ(30,520) MED1
    REWIND RRUNIT
    GO TO 105

C
C      MED2
550 M = NATCHR(CARD,COL)
    IF (M.EQ. BLANK) GO TO 105
    READ(30,520) MED2
    REWIND RRUNIT
    GO TO 105

C
C      *END*
550 CONTINUE

```

```

SET03810
SET03820
SET03830
SET03840
SET03850
SET03860
SET03870
SET03880
SET03890
SET03900
SET03910
SET03920
SET03930
SET03940
SET03950
SET03960
SET03970
SET03980
SET03990
SET04000
SET04010
SET04020
SET04030
SET04040
SET04050
SET04060
SET04070
SET04080
SET04090
SET04100
SET04110
SET04120
SET04130
SET04140
SET04150
SET04160
SET04170
SET04180
SET04190
SET04200
SET04210
SET04220
SET04230
SET04240
SET04250
SET04260
SET04270
SET04280
SET04290
SET04300
SET04310
SET04320
SET04330
SET04340
SET04350
SET04360
SET04370
SET04380
SET04390
SET04400
SET04410
SET04420
SET04430
SET04440
SET04450
SET04460
SET04470
SET04480
SET04490
SET04500
SET04510
SET04520
SET04530
SET04540
SET04550
SET04560

```

FILE SET14

```

C
C      PROCEDURE 1 AND 2 CANNOT BE SELECTED ALONG WITH PROCEDURE 3
C      IF (PROC.EQ. 3) GO TO 570
C      IF (NOFET2.EQ. 0 OR NOFEAT.EQ. 0) GO TO 570
C      IF (NOFET2.EQ. NOFEAT) GO TO 570
C      WRITE(6,2050)
C 570 IF (MANDOT.EQ. 0) GO TO 580
C      IF (PROC.EQ. 3) GO TO 580
C      WRITE(6,2060)
C      PROC. = 3
C 580 IF (MANSTA.EQ. 0) GO TO 590
C      IF (PROC.EQ. 3) GO TO 590
C      WRITE(6,2060)
C      PROC. = 3
C 590 CONTINUE
C      WRITE(6,1000)
C      IF (NOFET2.NE. 0) WRITE(6,1010) (FETVC2(I),I=1,NOFET2)
C      IF (NOFEAT.NE. 0) WRITE(6,1020) (FETVEC(I),I=1,NOFEAT)
C      IF (MAPKEY.EQ. 1) WRITE(6,1030)
C      IF (MANDOT.EQ. 1) WRITE(6,1040)
C      IF (MANSTA.EQ. 1) WRITE(6,1050)
C      IF (DSPKEY.EQ. 1) WRITE(6,1055)
C      IF (PROC.EQ. 1) WRITE(6,1060)
C      IF (PROC.EQ. 2) WRITE(6,1070)
C      IF (DIST.EQ. 1) WRITE(6,1080)
C      IF (DIST.EQ. 2) WRITE(6,1090)
C      WRITE(6,2000)
C      WRITE(6,2010) NFAWST
C      IF (SUNANG.EQ. 1) WRITE(6,2020)
C      IF (SUNANG.EQ. 2) WRITE(6,2030)
C      IF (SUNANG.EQ. 0) WRITE(6,2040)
C      IF (DOTKEY.EQ. 1) WRITE(6,2070)
C      IF (STATKY.EQ. 1) WRITE(6,2080)
C      IF (COND.EQ. 1) WRITE(6,2090)
C      IF (MIX.EQ. 1) WRITE(6,3000)
C      IF (PRNDOT.EQ. 1) WRITE(6,3020)
C      IF (PRNSTS.EQ. 1) WRITE(6,3030)
C      IF (EXIT.EQ. 1) WRITE(6,3040)
C
C      IERR = 0
C      IF (PROC.EQ. 3) GO TO 600
C      IF (STATKY.NE. 1) IERR = 1
C      IF (DOTKEY.NE. 1) IERR = 1
C      IF (COND.EQ. 1 OR MIX.EQ. 1) GO TO 594
C 593 IF (PROC.EQ. 1) GO TO 595
C 594 IF (MAPKEY.NE. 1) IERR = 1
C 595 IF (IERR.EQ. 1) WRITE(6,3010)
C 600 CONTINUE
C
C 1000 FORMAT(/' USER HAS REQUESTED THE FOLLOWING OPTIONS :'/)
C 1010 FORMAT(' STAT CHANNELS ARE',30(12,1X))
C 1020 FORMAT(' DOTDATA CHANNELS ARE',30(12,1X))
C 1030 FORMAT(' CLUSTER/CLASSIFICATION TAPE IS BEING INPUT')
C 1040 FORMAT(' DOTFIL WILL BE RELABELED')
C 1050 FORMAT(' STAT FILE WILL BE RELABELED')
C 1055 FORMAT(' MAPTAP FILE WILL BE OUTPUT')
C 1060 FORMAT(' K-NEAREST PROCEDURE WILL BE USED')
C 1070 FORMAT(' ALL-OF-A-KIND PROCEDURE WILL BE USED')
C 1080 FORMAT(' L1 DISTANCE WILL BE USED')
C 1090 FORMAT(' L2 DISTANCE WILL BE USED')
C 2000 FORMAT(' THRESHOLD DISTANCE = ',F10.3)
C 2010 FORMAT(' 1X,13,1-NEAREST DOTS WILL BE USED')
C 2020 FORMAT(' SUN ANGLES WILL BE EXTRACTED FROM DOTFIL')
C 2030 FORMAT(' SUN ANGLES WILL BE READ IN FROM CARDS')
C 2040 FORMAT(' NO SUN ANGLE CORRECTION WILL BE APPLIED')
C 2050 FORMAT(' NO. OF STAT CHANNEL AND DOT DATA CHANNELS MUST BE EQUAL')
C
C 2060 FORMAT(' A LABELING PROCEDURE MAY NOT BE CHOSEN WHEN UPDATING THE')
C 2070 FORMAT(' DOTFIL OR SAVTAP FILES')
C 2080 FORMAT(' DOTFIL FILE IS BEING INPUT')
C 2090 FORMAT(' SAVTAP FILE IS BEING INPUT')
C 2090 FORMAT(' CONDITIONAL CLUSTER MAP WILL BE OUTPUT')

```

SET04570
 SET04580
 SET04590
 SET04600
 SET04610
 SET04620
 SET04630
 SET04640
 SET04650
 SET04660
 SET04670
 SET04680
 SET04690
 SET04700
 SET04710
 SET04720
 SET04730
 SET04740
 SET04750
 SET04760
 SET04770
 SET04780
 SET04790
 SET04800
 SET04810
 SET04820
 SET04830
 SET04840
 SET04850
 SET04860
 SET04870
 SET04880
 SET04890
 SET04900
 SET04910
 SET04920
 SET04930
 SET04940
 SET04950
 SET04960
 SET04970
 SET04980
 SET04990
 SET05000
 SET05010
 SET05020
 SET05030
 SET05040
 SET05050
 SET05060
 SET05070
 SET05080
 SET05090
 SET05100
 SET05110
 SET05120
 SET05130
 SET05140
 SET05150
 SET05160
 SET05170
 SET05180
 SET05190
 SET05200
 SET05210
 SET05220
 SET05230
 SET05240
 SET05250
 SET05260
 SET05270
 SET05280
 SET05290
 SET05300
 SET05310
 SET05320

ORIGINAL PAGE IS
OF POOR QUALITY

FILE SET14

```

3000 FORMAT(' MIXED CLUSTER MAP WILL BE OUTPUT')
3010 FORMAT('/' USER HAS NOT INPUT ONE OF THE REQUIRED FILES: 'T20,
      ' SAVTAP MAPFIL OR DOTFIL')
3020 FORMAT(' PRINT UPDATED DOTFILE')
3030 FORMAT(' PRINT MEANS AND COVARIANCES')
3040  C  FORMAT(' EXIT IF INPUT LABEL NOT USED')
      C  RETURN
      C  END

```

```

SET05330
SET05340
SET05350
SET05360
SET05370
SET05380
SET05390
SET05400
SET05410
SET05420

```

FILE: STOMAP

```

C*      SURROUTINE STOMAP(ILINE,NSAMP,HIST,LIMIT,BEGIN)
C*      STODAT READS AND STORES THE CLASSIFICATION/CLUSTER MAP ON DRUM
C*      IMPLICIT INTEGER (A-Z)
C      INCLUDE COMRAK.LIST
COMMON/GLOBAL/HEAD(63),MAPTAP,DATAP,SAVTAP,BMFILE,BMKEY,
*      HISFIL,HISKEY,TRFORM,ERHPTP,ERPKEY,MAPUNT,NOFILE,
*      DRUMAD,DRUMDS,PAGSIZ,DATFIL,STAFIL,ASAV,ASAVFL
*      .NHSTUN,NHSTFI,SCTRUN,MAPFIL
*      .DOTUNT,DOTFIL,NCHPAS,TRNSFL,BMTREF,HISTFL,PCHUNT,
*      CRDUNT,PRUNT,RANDIO
CSEND  DIMENSION HIST(LIMIT),FETVEC(1),FLD(6),NLINE(4)
      TOTWRD = ILINE*NSAMP
      IF (TOTWRD .LE. (DRUMDS-(DRUMAD-BEGIN))) GO TO 120
      WRITE(6,110)
110     FORMAT(' NOT ENOUGH DRUM SPACE TO STORE DAS TAPE DATA')
      CALL CMERR
C      120 CALL TAPHDR(MAPUNT,MAPFIL)
      FETVEC(1) = 1
      NOFEAT = 1
      FLD(1) = 1
      FLD(2) = ILINE
      FLD(3) = 1
      FLD(4) = 1
      FLD(5) = NSAMP
      FLD(6) = 1
      REGIN = REGIN1
      CALL FLDINI(FLD ,FETVEC,NOFEAT)
      DUMPS = TOTWRD / LIMIT
      IF (MOD(TOTWRD,LIMIT) .NE. 0) DUMPS = DUMPS + 1
      TOTLNS = LIMIT / NSAMP
      IF (TOTLNS .GE. ILINE) GO TO 140
      DMP = DUMPS - 1
      DO 130 I=1,DMP
130     NLINE(1) = TOTLNS
      NLINE(DUMPS) = ILINE - TOTLNS*DMP
      GO TO 150
140     NLINE(1) = ILINE
C      150 DO 200 J=1,DUMPS
      NUMLIN = NLINE(J)
      DO 160 K=1,NUMLIN
      WORDS = NSAMP*(K-1)
160     CALL LINERD(HIST(WORDS+1),ENDTAP)
C      STORE ON HIGH SPEED DRUM
C      NWORDS = WORDS + NSAMP
      CALL RWRITE(REGIN,HIST(1),NWORDS,(STAT)
200     REGIN = REGIN + NLINE(J) * NSAMP
C      MAPFIL = MAPFIL + 1
C      RETURN
      END

```

ST000010
 ST000020
 ST000030
 ST000040
 ST000050
 ST000060
 ST000070
 ST000080
 ST000090
 ST000100
 ST000110
 ST000120
 ST000130
 ST000140
 ST000150
 ST000160
 ST000170
 ST000180
 ST000190
 ST000200
 ST000210
 ST000220
 ST000230
 ST000240
 ST000250
 ST000260
 ST000270
 ST000280
 ST000290
 ST000300
 ST000310
 ST000320
 ST000330
 ST000340
 ST000350
 ST000360
 ST000370
 ST000380
 ST000390
 ST000400
 ST000410
 ST000420
 ST000430
 ST000440
 ST000450
 ST000460
 ST000470
 ST000480
 ST000490
 ST000500
 ST000510
 ST000520
 ST000530
 ST000540
 ST000550
 ST000560
 ST000570
 ST000580

19. UTILITY SUBPROGRAMS

FILE: BMFIL

```

SUBROUTINE RMAT (RMAT,LCOMB,NOFET,VEC,KEY)
IMPLICIT INTEGER(A-Z)
REAL RMAT
C
C
C DEPENDING ON THE VALUE OF 'KEY', THIS SUBROUTINE PERFORMS ONE OF
C FIVE I/O FUNCTIONS, WITH REGARD TO THE TRANSFORMATION MATRIX 'B'
C
C KEY=1 B-MATRIX IS READ FROM CARDS AND STORED ON FILE.
C STORAGE MUST BE PROVIDED IN BMAT ARGUMENT
C KEY=2 B-MATRIX IS READ FROM FILE.
C KEY=3 THE VALUES OF LCOMB,NOFET AND VEC ARE READ FROM FILE.
C THIS FUNCTION CAN BE USED FOR ESTABLISHING DIMENSIONS
C FOR THE B-MATRIX.
C KEY=4 THE B-MATRIX IS PUNCHED ON CARDS.
C KEY=5 THE B-MATRIX IS WRITTEN ON FILE.
C
C *****
C ARGUMENTS: DEFINITION INPUT FOR OUTPUT FOR
C BMAT LINEAR TRANSFORMATION MATRIX KEY=4,5 KEY=1,2
C LCOMB NO. OF LINEAR COMBINATIONS KEY=4,5 KEY=1,2,3
C NOFET NO. OF FEATURES KEY=4,5 KEY=1,2,3
C VEC VECTOR CONTAINING FEATURES KEY=4,5 KEY=1,2,3
C USED IN OBTAINING B-MATRIX.
C DIMENSION-NOFET
C *****
C DIMENSION HMAT(1),VEC(1)
C INCLUDE COMMON/LIST
COMMON/GLOBAL/HEAD(63),MAPTAP,DATAP,SAPTAP,BMFILE,BMKEY,
* HISTFIL,HISKEY,TRFORM,ERIPTP,ERPKEY,MAPUNT,NOFILE,
* DRUMAD,DRMWDS,PAGSIZ,DATFIL,STAFIL,ASAV,ASAVFL
* ,NHSTUN,NHSTFI,SCRUN,MAFFIL
* ,DOTUNT,DOTFIL,NCHPAS,TRNSFL,BMTRFL,HISTFL,PCHUNT,
* CRDUNT,PRUNT,RANDIO
C
CSEND
GO TO((10,20,30,40),KEY)
10 READ(21,100)LCOMB,NOFET,(VEC(I),I=1,NOFET)
IK=LCOMB*NOFET
READ(21,200)(BMAT(I),I=1,IK)
GO TO 40
20 REWIND BMFILE
READ(BMFILE)LCOMB,NOFET,(VEC(I),I=1,NOFET)
IF(KEY.EQ.3)RETURN
IK=LCOMB*NOFET
READ(BMFILE)(BMAT(I),I=1,IK)
RETURN
30 WRITE(PCHUNT,300)
WRITE(PCHUNT,400)LCOMB,NOFET,(VEC(I),I=1,NOFET)
IK=LCOMB*NOFET
WRITE(PCHUNT,500)(BMAT(I),I=1,IK)
RETURN
40 REWIND BMFILE
WRITE(BMFILE)LCOMB,NOFET,(VEC(I),I=1,NOFET)
IK=LCOMB*NOFET
WRITE(BMFILE)(BMAT(I),I=1,IK)
RETURN
100 FORMAT(5X,I2,5X,I2,3X,30I2)
200 FORMAT(5X,5F15,6)
300 FORMAT('B-MATRIX CARDS')
400 FORMAT('LCOMB ',I2,'FEAT ',I2,'VEC ',30I2)
500 FORMAT('BMTRX ',5E15,8)
END

```

FILE: RNI4A1

```

C      SUBROUTINE RNI4A1(IFLD,INCHR,IBN)
C      DAVID LEE SMITH 17 OCTOBER 1977.
C      A SUBROUTINE TO CONVERT INTERNAL BINARY NUMBERS FROM
C      THE INTEGER*4 FORM TO A STRING OF EBCDIC CHARACTERS.
C
C      THE PARAMETERS ARE:
C      IFLD IS THE FIRST WORD OF A FIELD IN AN ARRAY IN WHICH
C      TO STORE THE OUTPUT EBCDIC CHARACTERS, ONE PER
C      WORD, IN A1 FORM, I.E., ONE CHARACTER PER WORD, WITH
C      BLANK FILL TO THE RIGHT.
C      INCHR IS THE NUMBER OF EBCDIC CHARACTERS = THE NUMBER OF
C      WORDS = THE WIDTH OF THE FIELD TO BE FILLED.
C      INCHR SHOULD ALSO BE OF FORM INTEGER*4.
C      IBN IS THE INPUT TO THIS ROUTINE, AN INTEGER IN INTERNAL
C      BINARY FORM, POSITIVE OR NEGATIVE (TWO'S COMPLEMENT)
C      IN THE LEGAL RANGE -2**31 TO (2**31-1).
C
C      INTEGER*4 IFLD(20), INCHR, IBN, IDIG(10), IRL, IHMI
C      DATA IDIG / 1H0, 1H1, 1H2, 1H3, 1H4, 1H5, 1H6, 1H7, 1H8, 1H9 /
C      DATA IHMI / 1H- /
C      DATA IRL / 1H8 /
C      LAST = INCHR
C      LBN = IBN
C      NEG = 0
C      IF ( LBN ) 10,30,30
C10  NEG = 1
C      LBN = -LBN
C20  IF ( LBN ) 20,30,30
C      IFLD(LAST) = IRL
C      LBN = 214 748 364
C      LAST = LAST - 1
C30  IF (LAST.LE. 0 ) GO TO 60
C      IPT = LAST
C      DO 50 I = 1, LAST
C      INDEX = MOD( LBN, 10 )
C      IFLD(IPT) = IDIG( INDEX + 1 )
C      LBN = LBN / 10
C      IPT = IPT - 1
C50  CONTINUE
C      IF ( NEG.NE. 0 ) IFLD(1) = IHMI
C60  RETURN
C      END

```

BNI00010
 BNI00020
 BNI00030
 BNI00040
 BNI00050
 BNI00060
 BNI00070
 BNI00080
 BNI00090
 BNI00100
 BNI00110
 BNI00120
 BNI00130
 BNI00140
 BNI00150
 BNI00160
 BNI00170
 BNI00180
 BNI00190
 BNI00200
 BNI00210
 BNI00220
 BNI00230
 BNI00240
 BNI00250
 BNI00260
 BNI00270
 BNI00280
 BNI00290
 BNI00300
 BNI00310
 BNI00320
 BNI00330
 BNI00340
 BNI00350
 BNI00360
 BNI00370
 BNI00380
 BNI00390
 BNI00400
 BNI00410

FILE: BUFILL

| | | |
|-----|---|----------|
| | SUBROUTINE BUFILL(IREC,IUNIT,MAXREC,IBUF,NRPDS,ENDTAP,IERR) | BUF00010 |
| | IMPLICIT INTEGER (A-Z) | BUF00020 |
| C | BUFILL READS THE MSS TAPE ONE RECORD AT A TIME. A MAXIMUM OF 10 | BUF00030 |
| C | RECORDS PER DATA SET MAY BE PROCESSED AT A TIME | BUF00040 |
| C | DIMENSION IBUF(765) | BUF00050 |
| | IERR = 0 | BUF00060 |
| | ENDTAP = 0 | BUF00070 |
| | K = 0 | BUF00080 |
| | DO 20 I=1,10 | BUF00090 |
| C | MAX = MAXREC / 4 | BUF00100 |
| | READ(IUNIT,100,ERR=50,END=60) (IBUF(K+J),J=1,MAX) | BUF00110 |
| 100 | FORMAT(31(250A4)) | BUF00120 |
| | IREC = IREC + 1 | BUF00130 |
| | IF (IREC .GE. NRPDS) RETURN | BUF00140 |
| | K = K + MAXREC/4 | BUF00150 |
| C | 20 CONTINUE | BUF00160 |
| | RETURN | BUF00170 |
| C | 50 WRITE(6,110) MAXREC,J | BUF00180 |
| 110 | FORMAT(1X,I4,' BYTES EXPECTED' / 1X,I4, ' BYTES ON RECORD') | BUF00190 |
| | IERR = -1 | BUF00200 |
| | RETURN | BUF00210 |
| C | ENCOUNTERED AN E-O-F | BUF00220 |
| C | 60 ENDTAP = -1 | BUF00230 |
| | RETURN | BUF00240 |
| C | END | BUF00250 |
| | | BUF00260 |
| | | BUF00270 |
| | | BUF00280 |
| | | BUF00290 |
| | | BUF00300 |
| | | BUF00310 |
| | | BUF00320 |
| | | BUF00330 |
| | | BUF00340 |
| | | BUF00350 |

ORIGINAL PAGE 1
OF POOR QUALITY

FILE: CHAIN

```

SUBROUTINE CHAIN(CLD)
C*****
C*
C* THIS SUBROUTINE CHAINS ALL CLUSTERS WHOSE MEANS ARE LESS THAN
C* DLMIN UNITS APART.
C* IF - DISTANCE BETWEEN CLUSTERS L AND M * DLMIN
C*       DISTANCE BETWEEN CLUSTERS L AND N = DLMIN
C*       DISTANCE BETWEEN CLUSTERS M AND N * DLMIN
C* THEN CLUSTERS L,M, AND N ARE CHAINED
C*
C* INPUT  CLD-CLUSTER DISTANCES
C*         DLMIN-MINIMUM DISTANCE BETWEEN CLUSTERS
C*         LNCAT-NUMBER OF CLUSTERS
C*
C* OUTPUT ICHAIN-ARRAY CONTAINING NUMBERS OF CHAINED CLUSTERS
C*         PRINTED SUMMARY OF CLUSTERS WHICH WERE CHAINED
C*****
C* IMPLICIT INTEGER (A-X)
C* INCLUDE COMRKS.LIST
C* INCLUDE COMRKA.LIST
C* COMMON/PASS/STOP,LNCAT,NMIN,KRN,STDMAX,DLMIN,SEP,
C*       MAP,SPTRIG, IRD, KPTS, NOPTS, PUNCH,
C*       ICHN,CHNTHS,ICHAIN(62),NWDS,IREGIN,BEGIN1,
C*       BEGIN2,BEGIN3,CLSNAM,NOFLD,IPT,TOTWRD,TOTPTS,
C*       NCLASS,NOCLS,TOTSUR,TOTFLO,TOTVRT,NOCL,NVRT
C*       ,NXTCLS,NOFEAT,MAXCLS,FETVFC(30),SYMMTX(62)
C*       ,VARSI7,STATKY,ISOKEY,MAPFMT,MAPKEY,SEQUEN(20),PERCEN,SIMERP
C*       ,IORDER,INUNIT,INFILE,INITM,PMIN,SUBVEC(62),NOSUB2,CHNVC(30)
C*       ,NOCHAN,FRCOMP,NOSEQ,MEANDQ,MEANDU,
C*       ,SYMDQ,SYMDU,ITRIGQ,ITRIGU,DOFLAG,
C*       ,DUFLAG,DOOU,STDOTS(60),NSDOTS,SUNCOR(30),LLNCAT,
C*       ,DVERT(250,2),DRECT(60,2),DVPNT(11,2),IDCNT(2),NDOU(2)
C*       ,MXFET1,MAXPOP
C* REAL SUNCOR
C* COMMON/GLOBAL/HEAD(63),MAPTAP,DATAPE,SAVTAP,BMFILE,BMKEY,
C*       HISFIL,HISKEY,TRFORM,ERIPTP,ERPKEY,MAPUNT,NOFILE,
C*       ,DRUMAD,DRMADS,PAGSIZ,DATFIL,STAFIL,ASAV,ASAVFL
C*       ,NHSTUN,NHSTFI,SCRUN,MAPFIL
C*       ,DOTUNT,DOTFIL,NCHPAS,TRNSFL,BMTRFL,HISTFL,PCHUNT,
C*       ,CROUNT,RTUNT,RANDIO
C*END
EQUIVALENCE (SYMBLS,SYMMTX)
DIMENSION JP(62),CLD(MAXCLS,MAXCLS),SYMBLS(62)
REAL CHNTHS,CLD
IHD=0
DO 10 I=1,LNCAT
10 ICHAIN(I) = 1
20 DO 30 J=1,LNCAT
30 JP(I) = ICHAIN(I)
I=0
40 I=I+1
IF (I.GE.LNCAT) GO TO 60
M=I+1
DO 50 J=M,LNCAT
IF (CLD(I,J).GT.CHNTHS) GO TO 50
ICHAIN(I) = MIN0(ICHAIN(I),ICHAIN(J))
ICHAIN(J) = ICHAIN(I)
50 CONTINUE
GO TO 40
60 DO 70 I=1,LNCAT
IF (ICHAIN(I).NE.JP(I)) GO TO 20
70 CONTINUE
M=1
KNCAT=LNCAT
80 K=0
IM=M+1
DO 90 I=IM,LNCAT
IF (ICHAIN(I).NE.M) GO TO 90
KNCAT=KNCAT+1
K=K+1
SYMBLS(I)=SYMBLS(M)
JP(K)=I
90 CONTINUE
IF (K.EQ.0) GO TO 100
IF (IHD.EQ.0) WRITE (6,140)
IF (IHD.EQ.0) WRITE (6,HEAD)
IHD=1
WRITE (6,110)M,(JP(I),I=1,K)

```


FILE: CHAIN

```
100 WRITE (6,120)M
    M=M+1
    IF (M.LT.LNCAT) GO TO 80
    IF (KNCAT.EQ.LNCAT) RETURN
    WRITE (6,130)KNCAT
    RETURN
110 FORMAT(/' THE FOLLOWING CLUSTERS SHOULD BE CHAINED---',20I4)
120 FORMAT(/' IN THE FINAL OUTPUT MAP ALL OF THE ABOVE CLUSTERS WILL BE
    *F REPRESENTED BY THE SYMBOL FOR CLUSTER',I4//)
130 FORMAT(' THE ABOVE CHAINING REDUCES THE EFFECTIVE NUMBER OF CLUSTERS TO ',I5)
140 FORMAT(1H1)
    END
```

CHA00510
CHA00620
CHA00630
CHA00640
CHA00650
CHA00660
CHA00670
CHA00680
CHA00690
CHA00700
CHA00710
CHA00720
CHA00730

ORIGINAL PAGE IS
OF POOR QUALITY

FILE CHLDET

```

SUBROUTINE CHLDET( KKK,NV,DUM,DET)
    THIS ROUTINE COMPUTES THE MODIFIED CHOLESKY DECOMPOSITION OF
    THE COVARIANCE MATRIX. THE DECOMPOSITIONS OVERLAY THE ELEMENTS
    OF THE COVARIANCE MATRIX.
    KK = L D L*
    KK = COVARIANCE MATRIX STORED IN SYMMETRIC STORAGE
    NV = NO. OF CHANNELS
    DUM = A WORK AREA OF SIZE NV-1
    DET = THE DETERMINANT OF THE COVARIANCE MATRIX

    REAL KK, KKK
    LOGICAL JEL
    DIMENSION KKK(1), DUM(1), KK(465)

    COPY COVARIANCE MATRIX FROM KKK TO KK TO AVOID OVERSTORING
    THE INPUT MATRIX
    ISIZE = (NV*(NV+1))/2
    DO 5 I=1, ISIZE
        KK(I) = KKK(I)
    CONTINUE

    DOUBLE PRECISION TF, R, R1, DUM, T1
    JEL = .TRUE.
    J1 = 0
    JD = 0
    DET = 1.0

    LOOP OVER ALL CHANNELS
    DO 10 J=1, NV
        KL = J-1
        L = J+1
        JD = J1
        J1 = J1 + J
        TF = KK(J1)
        IF(JEL) GO TO 12
        K1 = 0

        COMPUTE THE DIAGONAL ELEMENTS OF D AND STORE IN KK
        TEMPORARILY STORE THE PRODUCT KK(I,I)*KK(J,I) IN DUM(I)
        DO 15 I=1, KL
            R = KK(JD + I)
            K1 = K1 + I
            R1 = KK(K1) * R
            TF = TF - R1 * R
            DUM(I) = R1
        15 CONTINUE
        KK(J1) = TF
        12 CONTINUE
        DET = DET * TF
        IF (L .GT. NV) GO TO 10
        IRD = J1 - L + 1

        COMPUTE THE R, J-TH ELEMENT OF L, USING T1
        DO 20 IR=L, NV
            IRD = IRD + IR - 1
            T1 = KK(IRD + J)
            IF(JEL) GO TO 16
            DO 25 I=1, KL
                T1 = T1 - DUM(I) * KK(IRD + I)
            25 CONTINUE
            IF (TF.GT.0.D0) GO TO 17
            DET=0
        16
    
```

MCH00010
 MCH00020
 MCH00030
 MCH00040
 MCH00050
 MCH00060
 MCH00070
 MCH00080
 MCH00090
 MCH00100
 MCH00110
 MCH00120
 MCH00130
 MCH00140
 MCH00150
 MCH00160
 MCH00170
 MCH00180
 MCH00190
 MCH00200
 MCH00210
 MCH00220
 MCH00230
 MCH00240
 MCH00250
 MCH00260
 MCH00270
 MCH00280
 MCH00290
 MCH00300
 MCH00310
 MCH00320
 MCH00330
 MCH00340
 MCH00350
 MCH00360
 MCH00370
 MCH00380
 MCH00390
 MCH00400
 MCH00410
 MCH00420
 MCH00430
 MCH00440
 MCH00450
 MCH00460
 MCH00470
 MCH00480
 MCH00490
 MCH00500
 MCH00510
 MCH00520
 MCH00530
 MCH00540
 MCH00550
 MCH00560
 MCH00570
 MCH00580
 MCH00590
 MCH00600
 MCH00610
 MCH00620
 MCH00630
 MCH00640
 MCH00650
 MCH00660
 MCH00670
 MCH00680
 MCH00690
 MCH00700
 MCH00710
 MCH00720
 MCH00730
 MCH00740
 MCH00750
 MCH00760

FILE CHLDET

| | | |
|-----------------------|---|----------|
| | RETURN | MCH00770 |
| 17 | KK(IRO * J) = T1/TF | MCH00780 |
| 20 | CONTINUE | MCH00790 |
| | JF1 = .FALSE. | MCH00800 |
| 10 | CONTINUE | MCH00810 |
| C
C
C
C
C | KK CONTAINS . IN 'SYMMETRIC' STORAGE, THE MODIFIED CHOLESKY
FACTORIZATION OF THE INPUT MATRIX. THE LOWER TRIANGULAR MATRIX, L,
OCCUPIES THE OFF-DIAGONAL ELEMENTS OF KK . AND THE DIAGONAL
MATRIX, D . IS STORED IN THE DIAGONAL ELEMENTS IN KK. | MCH00820 |
| | | MCH00830 |
| | | MCH00840 |
| | | MCH00850 |
| | | MCH00860 |
| | RETURN | MCH00870 |
| | END | MCH00880 |
| | | MCH00890 |

ORIGINAL PAGE IS
OF POOR QUALITY

```

C *****
C THIS SUBROUTINE CALCULATES THE WEIGHTED DISTANCE BETWEEN
C CLUSTER MEANS
C
C INPUT      AMN(MEANS) - MEANS OF EACH FEATURE OF EACH CLUSTER
C             STDEV      - STANDARD DEVIATIONS FOR EACH FEATURE/CLUSTER
C             LNCA       - NUMBER OF CLUSTERS
C             NOFEAT      - NUMBER OF FEATURES (CHANNELS)
C
C OUTPUT     CLD         - ARRAY CONTAINING DISTANCE BETWEEN CLUSTERS
C                   CLD(N,M)=DISTANCE BETWEEN CLUSTERS N AND M
C *****
C
C SURROUTINE CLDIST(CLD,STDEV,MEANS)
C IMPLICIT INTEGER (A-Z)
C INCLUDE COMBKS.LIST
C COMMON/PASS/STOP,LNCA,NMIN,KRN,STDMAX,DLMIN,SEP,
C * MAP,SPTRIG, IWD, KPIS, NOPTS, PUNCH,
C * ICHN,CHNTHS,ICHAIN(62),NWDS,IHGIN,REGIN1,
C * REGIN2,REGIN3,CLSNAM,NOFLD,IPT,TOTWRD,TOTPTS,
C * NCLASS,NOCLS,TOTSUR,TOTFLD,TOTVRT,NOCL,NVRT
C * ,NXTCLS,NOFEAT,MAXCLS,FETVEC(30),SYMMTX(62)
C * ,VARS17,STATKY,ISOKEY,MAPFMT,MAPKEY,SEQUEN(20),PERCEN,SIMERP
C * ,IORDER,INUNIT,INFILE,INITH,PMIN,SUBVEC(62),NOSUB2,CHNVC(30)
C * ,NOCHAN,FRCOMP,NOSEQ,MEANDU,MEANDU,
C * SYMDU,SYMDU,ITRIG0,ITRIGU,DOFLAG,
C * DUFLAG,ODDU,STDOTS(60),NSDOTS,SUNCOR(30),LLNCA,
C * DVRT(250,2),DRECT(60,2),DVPNT(11,2),IDCNT(2),NDOU(2)
C * ,MAXFT1,MAXPOP
C REAL SUNCOR
C
C REAL STDEV,MEANS,CLD
C DIMENSION MEANS(NOFEAT,MAXCLS)
C DIMENSION CLD(MAXCLS,MAXCLS),STDEV(NOFEAT,MAXCLS)
C DO 30 I=1,LLNCA
C   CLD(I,I)=0.
C   JJ=1+1
C   IF (I.EQ.LLNCA) JJ=I
C   DO 20 J=JJ,LLNCA
C     CLD(I,J)=0.
C     DO 10 K=1,NOFEAT
C       IF (STDEV(K,I).GT.0.0 .AND. STDEV(K,J).GT.0.0) GO TO 5
C       CLD(I,J)=999.99
C     GO TO 15
C   5 CONTINUE
C     CLD(I,J)=CLD(I,J)+(MEANS(K,I)-MEANS(K,J))**2/(STDEV(K,I)*STDEV(K,J)
C     *J)
C   10 CONTINUE
C     CLD(I,J)=SORT(CLD(I,J))
C   15 CLD(J,I) = CLD(I,J)
C   20 CONTINUE
C   30 RETURN
C END

```

ORIGINAL PAGE IS
OF POOR QUALITY

FILE: CLSCHK

```

      SURROUTINE CLSCHK (CLSDS, SUBDES, FLDSAV, VERTEX, SUBNO,
      * NOFEAT, FETVEC, NOCLS, NOFLD, BMFLG, NOSUB)
C
C      IMPLICIT INTEGER (A-H, O-Z)
C-----
C      THIS SURROUTINE IS CALLED FROM REDSAV TO CHECK THE VALIDITY OF
C      USER REQUESTS REGARDING SUBCLASSES, GROUPING AND CHANNELS
C-----
C      INCLUDE COMRK1.LIST
C      COMMON/INFORM/NOCLS2, NOSUR2, NOFET2, VARSZ2, TOTVT2, NOFLD2,
C      * AVAR2, COVAR2, CLSID2, SUBNO2, SURDS2, FLDSV2, VERTX2,
C      * FETVC2(30), SURVC2(75), SUBPTR(75), CLSVC2(60),
C      * KEPPT5(60), NOGRP, GRPNAM(60), GRPDEX(61),
C      * GRPCHK(61), GROUPS(124)
CSEND
C-----
C      DIMENSION INVERT(30)
C      DIMENSION SETI(60)
C      DATA MAXFET/30/
C      DIMENSION FETVEC(30)
C
C      DIMENSION CLSDS(1), SUBDES(1), FLDSAV(4, NOFLD), VERTEX(1)
C      DIMENSION SUBNO(1)
C-----
C      IF (NOSUB2.LE.0) GO TO 40
C      II = 0
C      IRIG = 0
C      DO 30 I=1, NOSUB2
C      J = SURVC2(I)
C      IF (J.LE.IRIG .OR. J.GT.NOSUB) GO TO 10
C      II = II+1
C      SURVC2(II)=J
C      IRIG = J
C      GO TO 30
10  WRITE (6,20) J
20  FORMAT (//5X, 'CLSCHK** - REQUESTED SUBCLASS NO.,', I3, ' IS NOT AVA
* ILABLE IN INPUT STATISTICS -- REQUEST IGNORED/')
30  CONTINUE
C      NOSUR2=II
C      IF (NOSUR2.GT.0) GO TO 60
40  NOSUR2=NOSUR
C      IF (NOSUR.GT.60) NOSUB2=60
C      DO 50 I=1, NOSUR2
50  SURVC2(I) = I
C-----
C      CHECK THE GROUPS FOR VALIDITY
C-----
60  GRPTR = 0
C      IF (NOGRP.LE.0) GO TO 110
C      II = 0
C      DO 100 I=1, NOGRP
C      JR = GRPDEX(I)+1
C      JE = JR+GROUPS(JR-1)-1
C      GRPTR = GRPTR+1
C      II = GRPTR
C      DO 90 J=JR, JE
C      JJ = GROUPS(J)
C      IF (JJ.GT.NOSUR) GO TO 70
C      GRPTR = GRPTR+1
C      GROUPS(GRPTR) = JJ
C      GO TO 90
70  WRITE (6,80) JJ, I
80  FORMAT (//5X, 'CLSCHK** - REQUESTED SUBCLASS NO.,', I3, ' FOR GROUP
* NO.,', I3, ' IS NOT AVAILABLE IN INPUT STATISTICS FILE/')
90  CONTINUE
C      GRPTR = GRPTR-1
C      IF (GRPTR.LT.II) GO TO 100

```

FILE: CLSCHK

```

      GRPTR = GRPTR+1
      GROUPS(I) = GRPTR-1
      II = II+1
      GRPDIX(I) = II
      GRPNAM(I) = GRPNAM(I)
100  CONTINUE
      NOGRP = II
C
C
C
      DELETE ALL GROUP SUBCLASSES
-----
110  IC=NOSUB2
      IF (NOGRP.LE.0) GO TO 170
      DO 140 I=1,NOGRP
      JR = GRPDIX(I)+1
      JE = JR+GROUPS(JR-1)-1
      DO 130 J=JB,JE
      KK = 0
      DO 120 K=1,NOSUB2
      IF (SUBVC2(K).EQ.GROUPS(J)) GO TO 120
      KK = KK+1
      SUBVC2(KK) = SUBVC2(K)
120  CONTINUE
130  NOSUB2 = KK
140  CONTINUE
C
C
C
      ADD BACK FIRST CLASS FROM EACH GROUP
-----
150  IC = NOSUB2
      DO 160 I=1,NOGRP
      NOSUB2 = NOSUB2+1
      WAT=GRPDIX(I)
      II=GROUPS(WAT+1)
      SUBVC2(II)=GRPNAM(I)
160  SUBVC2(NOSUB2)=II
C
C
C
      CONSTRUCT GROUP FOR EACH SUBCLASS NOT EXPLICITLY GROUPED
-----
170  IF (IC.LE.0) GO TO 190
      DO 180 I=1,IC
      NOGRP = NOGRP+1
      GRPTR = GRPTR+1
      GRPDIX(NOGRP) = GRPTR
      GROUPS(GRPTR) = 1
      II = SUBVC2(I)
      GRPNAM(NOGRP) = SUBDES(II)
      GRPTR = GRPTR+1
      GROUPS(GRPTR) = II
180  CONTINUE
C
C
C
      ARRANGE 'SUBVC2' IN ORDER
-----
190  CALL ORDER(SUBVC2,NOSUB2)
C*  USE SURPTR STORAGE TEMPORARILY TO SEE IF ANY ENTIRE CLASSES
C*  HAVE BEEN ELIMINATED.
      IK=0
      DO 191 I=1,NOCLS
      K=SURN0(I)
      SURN0(I)=0
      DO 191 L=1,K
      IK=IK+1
191  SURPTR(IK)=I
      KNT=0
      LC=1
      DO 194 I=1,NOSUB2
      IK=SUBVC2(I)
      IC=SURPTR(IK)
      IF (IC.EQ.LC) GO TO 193
      SURN0(LC)=KNT
      LC=IC
      KNT=1
      GO TO 194
193  KNT=KNT+1
194  CONTINUE
      SURN0(LC)=KNT
      DO 192 I=1,NOCLS

```

CLS00760
 CLS00770
 CLS00780
 CLS00790
 CLS00800
 CLS00810
 CLS00820
 CLS00830
 CLS00840
 CLS00850
 CLS00860
 CLS00870
 CLS00880
 CLS00890
 CLS00900
 CLS00910
 CLS00920
 CLS00930
 CLS00940
 CLS00950
 CLS00960
 CLS00970
 CLS00980
 CLS00990
 CLS01000
 CLS01010
 CLS01020
 CLS01030
 CLS01040
 CLS01050
 CLS01060
 CLS01070
 CLS01080
 CLS01090
 CLS01100
 CLS01110
 CLS01120
 CLS01130
 CLS01140
 CLS01150
 CLS01160
 CLS01170
 CLS01180
 CLS01190
 CLS01200
 CLS01210
 CLS01220
 CLS01230
 CLS01240
 CLS01250
 CLS01260
 CLS01270
 CLS01280
 CLS01290
 CLS01300
 CLS01310
 CLS01320
 CLS01330
 CLS01340
 CLS01350
 CLS01360
 CLS01370
 CLS01380
 CLS01390
 CLS01400
 CLS01410
 CLS01420
 CLS01430
 CLS01440
 CLS01450
 CLS01460
 CLS01470
 CLS01480
 CLS01490
 CLS01500
 CLS01510
 CLS01520
 CLS01530
 CLS01540

FILE: CLSCHK

```

GO TO 2A0
272 WRITE(6,273) (FETVC2(K),K=1,NOFET2)
273 FORMAT(// 5X,100000 B-MATRIX CHANNELS MUST BE EQUAL TO OR A SUB-CLASS
1ET OF AVAILABLE TRAINING DATA CHANNELS --- THE INPUT B-MATRIX CHANNEL
2NEL SET IS ...// (5X,3014//) )
WRITE(6,274)
274 FORMAT(// 5X,100000 TERMINATING PROGRAM EXECUTION FROM FETCHK
100000//)
CALL EXIT
2A0 CONTINUE
NOFET2 = 11
C
C
C SET UP REVISED INVERT TABLE
-----
DO 290 I=1,MAXFET
290 INVERT(I) = 0
DO 300 I=1,NOFET2
WAT=FETVC2(I)
300 INVERT(WAT) = I
C
C
C SET UP CLSVC2 ARRAY SO THAT IT CONTAINS THE CLASS NO. TO WHICH THE
C CORRESPONDING SURCLASS BELONGS.
J=0
DO 305 I=1,NOCLS2
IK=SURNO(I)
DO 305 K=1,IK
J=J+1
305 CLSVC2(J)=I
C
C
C SAVE FIELD DESCRIPTIONS FOR CLASSES AND SUBCLASSES TO BE USED.
NOFLD2=0
IV=0
JV=0
DO 330 I=1,NOFLD
JR=FLOSAV(3,I)
C ARE FIELDS ASSOCIATED WITH SUBCLASSES
IF(JR.NE.0)GO TO 306
IC=FLOSAV(2,I)
C HAS THIS CLASS BEEN ELIMINATED
NC=SETI(IC)
IF(NC.EQ.-1)GO TO 320
NSUR=0
GO TO 307
306 CONTINUE
C HAS THIS SURCLASS BEEN ELIMINATED
NSUR = SURPTR(JR)
IF(NSUR.EQ.0)GO TO 320
NC=CLSVC2(NSUR)
307 CONTINUE
NOFLD2=NOFLD2+1
FLOSAV(1,NOFLD2)=FLOSAV(1,I)
FLOSAV(2,NOFLD2)=NC
FLOSAV(3,NOFLD2)=NSUR
FLOSAV(4,NOFLD2)=FLOSAV(4,I)
C
C
C NOW SAVE VERTICES
NV=FLOSAV(4,I)*2
DO 310 J=1,NV
IV=IV+1
310 VERTEX(IV) = VERTEX(JV+J)
JV=JV+NV
GO TO 330
320 JV = JV + FLOSAV(4,I)*2
330 CONTINUE
TOTVT2=IV/2
RETURN
END

```

CLS02340
CLS02350
CLS02360
CLS02370
CLS02380
CLS02390
CLS02400
CLS02410
CLS02420
CLS02430
CLS02440
CLS02450
CLS02460
CLS02470
CLS02480
CLS02490
CLS02500
CLS02510
CLS02520
CLS02530
CLS02540
CLS02550
CLS02560
CLS02570
CLS02580
CLS02590
CLS02600
CLS02610
CLS02620
CLS02630
CLS02640
CLS02650
CLS02660
CLS02670
CLS02680
CLS02690
CLS02700
CLS02710
CLS02720
CLS02730
CLS02740
CLS02750
CLS02760
CLS02770
CLS02780
CLS02790
CLS02800
CLS02810
CLS02820
CLS02830
CLS02840
CLS02850
CLS02860
CLS02870
CLS02880
CLS02890
CLS02900
CLS02910
CLS02920
CLS02930
CLS02940
CLS02950
CLS02960
CLS02970
CLS02980
CLS02990
CLS03000
CLS03010
CLS03020

FILE: CLSHIS

```

C      SUBROUTINE CLSHIS(TALLY,HISBUF,TTL,XSIZ,XHGM,XLOW,YSIZ,
C      *NOMIST,FLOPTS,HISVEC)
C      IMPLICIT INTEGER (A-H,O-Z)
C      REAL XSCALE,XSHFT
C      INCLUDE COMRK6.LIST
C      COMMON/GLOBAL/HEAD(63),MAPTAP,DATAP,SAVTAP,BMFILE,BMKEY,
C      *HISFIL,HISKEY,THFORM,ER1PTP,ERPKEY,MAPUNT,NOFILE,
C      *DRUMAD,DRMUNS,PAGSIZ,DATFIL,STAFIL,ASAV,ASAVFL,
C      *NMSTUN,NMSTFI,SCTRUN,MAPFIL,
C      *DOTUNT,DOTFIL,NCHPAS,TRANSFL,BMTRFL,HISTFL,PCMUNT,
C      *CRDUNT,PRUNT,RANDIO
C      DIMENSION HISVEC(30)
C      DIMENSION TALLY(NOMIST,XSIZ),HISBUF(XSIZ),XAXIS(11)
C      LOGICAL*1 DUMM(4),SYMM(4)
C      EQUIVALENCE (SYM,SYMM(1)),(DUM,DUMM(1))
C      DATA STAR/'*'/, BLANK/' '/, ALPHA1/ZC0/
C      DATA DOLBCD/'S'/, NUMIC/ZF0/, ALPHA1/ZC0/
C      -----
C      GO = 1
C      INIZ
C      -----
10  JPTCNT = (PAGSIZ-R)/(YSIZ + 10 )
    NSIZ = (XSIZ+R)/10 + 1
    XSCALE = FLOAT(XLOW-XHGM)/(XSIZ-1)
    XSHFT = FLOAT(XSIZ*XHGM-XLOW)/(XSIZ-1)
    DO 20 I=1,NSIZ
20  XAXIS(I) = (10*I-9)*XSCALE + XSHFT + 0.50
C      JCNT = JPTCNT
    DO 160 JFEAT=1,NOMIST
    IF (JCNT.LT.JPTCNT) GO TO 40
    WRITE (6,HEAD)
    IF (GO,EQ.4) WRITE (6,11)
11  FORMAT(' DATA TR')
    IF (GO,EQ.1) WRITE (6,200) TTL
    IF (GO,EQ.0) WRITE (6,201) TTL
200  FORMAT('T63, HISTOGRAM',T59,'-----//T55, TRAINING SUBC
    *LASS',A4/T53,'-----//')
201  FORMAT('T63, HISTOGRAM',T59,'-----//T55, TRAINING FIEL
    *D',A4/T53,'-----//')
    IF (GO,EQ.2) WRITE (6,145) TTL, FLOPTS
    IF (GO,EQ.3) WRITE (6,147) TTL
    IF (GO,EQ.0) WRITE (6,30) FLOPTS, TITLE
30  FORMAT('T4R, (NO. SAMPLES= ',I7,' , SUBCLASS= ',A4,' )//')
    JCNT = 0
C      SCALE AND PRINT THE HISTOGRAM
C      -----
40  MAX = 0
    YSCALE = 1
    JCNT = JCNT + 1
    DO 50 J=1,XSIZ
    IF (TALLY(JFEAT,J) .GT. MAX) MAX = TALLY(JFEAT,J)
50  CONTINUE
    IF (MAX .GT. YSIZ) YSCALE = (MAX+(YSIZ-1))/YSIZ
    K = HISVEC(JFEAT)
    IF (GO,EQ.4) GO TO 62
    WRITE (6,60) K
60  FORMAT(' CHANNEL ',I2/1X,'-----')
62  CONTINUE
    WRITE (6,70) YSCALE
70  FORMAT('0',T5,'EACH * REPRESENTS',I4,' POINT(S).//')
C      DO 120 JY=1,YSIZ
    JH = (YSIZ-(JY-1))*YSCALE
    IK = JH - YSCALE
    JEMP=XSIZ

```

FILE: CLSHIS

```

DO 100 IAZ=1,JEMP
  I=IAZ
  HISBUF(I) = BLANK
  JK = TALLY(JFEAT,I)
  SYM = STAR
  IF (JK.GE.JH) GO TO 90
  IF (JK.LE.IK) GO TO 100
  JK = JK-IK
  ZONE = NUMIC
  IF (JK.LT.10) GO TO 80
  ZONE = ALPHA1
  JK = JK-9
  IF (JK.LT.10) GO TO 80
  SYM = DOLBCD
  GO TO 90
80  DUM = ZONE + JK
  SYMM(1)=DUMM(4)
90  HISBUF(I) = SYM
100 CONTINUE
  WRITE (6,110) JH, (HISBUF(I),I=1,XSIZ)
110 FORMAT(1X,16,' I',1X,112A1)
120 CONTINUE
  WRITE (6,130) (XAXIS(I),I=1,DSIZ)
130 FORMAT(10X,10('-----'),',',',',/9X,11(13,7X))
  WRITE (6,140)
140 FORMAT('0')
145 FORMAT(T63,'HISTOGRAM',/T59,18(1H-)//T61,'FIELD',',2X,A4,
  / T57, 22(1H-)// T56, '( NO. SAMPLES : ', 17,' )' )
147 FORMAT(T63,'HISTOGRAM',/T59,18(1H-)//T65,A4,
  / T57, 22(1H-))
  DO 150 I=1,XSIZ
150 TALLY(JFEAT,I) = 0
160 CONTINUE
  CALL SETMRG( 66,1,65 )
  RETURN
-----
C
C
C  ENTRY FLDHIS(TALLY,HISBUF,TTL,XSIZ,XHGH,XLOW,YSIZ,
C  *NOHIST,FLDPTS,TITLE,HISVEC)
C
C  GO = 0
C  GO TO 10
C
C  -----
C
C  ENTRY HSTGRM(TALLY,HISBUF,TTL,PRINT,XSIZ,XHGH,XLOW,YSIZ,
C  *NOHIST,FLDPTS,HISVEC)
C
C  GO = PRINT
C
C  GO TO 10
C
C  ENTRY COMHST(TALLY,HISBUF,TTL,NOHIST,HISVEC,XSIZ,XHGH,XLOW,YSIZ)
C  GO = 4
C  DO 12 I=1,NOHIST
C  DO 61 J=1,50
C  ITEM=TALLY(I,J)
C  JT=102-J
C  TALLY(I,J)=TALLY(I,JT)
61  TALLY(I,JT)=ITEM
12  CONTINUE
C  GO TO 10
C  END

```

CLS00800
 CLS00810
 CLS00820
 CLS00830
 CLS00840
 CLS00850
 CLS00860
 CLS00870
 CLS00880
 CLS00890
 CLS00900
 CLS00910
 CLS00920
 CLS00930
 CLS00940
 CLS00950
 CLS00960
 CLS00970
 CLS00980
 CLS00990
 CLS01000
 CLS01010
 CLS01020
 CLS01030
 CLS01040
 CLS01050
 CLS01060
 CLS01070
 CLS01080
 CLS01090
 CLS01100
 CLS01110
 CLS01120
 CLS01130
 CLS01140
 CLS01150
 CLS01160
 CLS01170
 CLS01180
 CLS01190
 CLS01200
 CLS01210
 CLS01220
 CLS01230
 CLS01240
 CLS01250
 CLS01260
 CLS01270
 CLS01280
 CLS01290
 CLS01300
 CLS01310
 CLS01320
 CLS01330
 CLS01340
 CLS01350
 CLS01360
 CLS01370
 CLS01380
 CLS01390
 CLS01400
 CLS01410
 CLS01420
 CLS01430

FILE: CMERR

```
SUBROUTINE CMERR
WRITE(6,100)
100 FORMAT(' ERROR HAS OCCURRED')
CALL EXIT
RETURN
END
```

CME00010
CME00020
CME00030
CME00040
CME00050
CME00060

```

SUBROUTINE CRDSTA (ARRAY, TOP)
IMPLICIT INTEGER (A-Z)*X)
INCLUDE COMCHK1.LIST
INCLUDE COMCHK6.LIST
COMMON/IN ORP/NOCLS2,NOSUR2,NOFET2,VARSZ2,TOTVRT2,NOFLD2,
* AVAR2,COVAR2,CLSID2,SUBNO2,SURDS2,FLDSV2,VERTX2,
* FETVC2(30),SURVC2(75),SURPTR(75),CL SVC2(60),
* KEPPTS(60),NOGRP,GRPNAM(60),GRPDEX(61),
* GRPCHK(61),GROUPS(124)
COMMON/GLOBAL/HEAD(63),MAPTAP,DATAPE,SAVTAP,BMFILE,BMKEY,
* HISFIL,HISKEY,TRFORM,ERIPTP,ERPKEY,MAPUNT,NOFILE,
* DRUMAD,DRM4DS,PAGSIZ,DATFIL,STAFIL,ASAV,ASAVFL
* ,NHSTUN,NHSTFI,SCRUN,MAPFIL
* ,DOTUNT,DOTFIL,NCHPAS,TRNSFL,BMTRFL,HISTFL,PCHUNT,
* CRDUNT,PRUNT,WANDIO
C$END
COMMON /PASSB/ NOCLS,NOSUB,NOFEAT,NOFLD,TOTVRT,
* FETVEC(30),FLDSV1,CLSID1,VARSIZ
DIMENSION APRAY(1)

READ FROM CARDS KEY WORDS TO BE USED IN BASE ADDRESSES
200 READ(CRDUNT,200)NOCLS,NOSUR,NOFEAT,NOFLD,TOTVRT
FORMAT(5X,I4,8X,I2,8X,I2,7X,I3,8X,I4)
210 READ(CRDUNT,210) (FETVEC(I),I=1,NOFEAT)
FORMAT(10X,30I2)

COMPUTE BASE ADDRESSES
VARSIZ = NOFEAT * (NOFEAT + 1) / 2
MAXFLD = NOFLD
COVAR1 = 1
AVAR1 = COVAR1 + VARSIZ
MAXCLS = NOSUB
CLSID1 = AVAR1 + NOFEAT
SUBNO1 = CLSID1 + NOCLS
SURDS1 = SUBNO1 + NOCLS
FLDSV1 = SURDS1 + NOSUB
VERTX1 = FLDSV1 + NOFLD*4
TIPTOP = VERTX1 + TOTVRT
RANCOR = TOP - TIPTOP
IF (BADCOR .LT. 0) GO TO 100

CALL RDMODK (ARRAY(AVAR1),ARRAY(COVAR1),ARRAY(CLSID1),ARRAY(SUBNO1),
* ,ARRAY(SURDS1),ARRAY(FLDSV1),ARRAY(VERTX1),ARRAY(1))

GO TO 90
100 WRITE (PRUNT,190)
190 FORMAT(' EXCEEDED CORE LIMITS. REDUCE NO. OF TRAINING CLASSES OR F
* FEATURES, / ' EXITING FROM CRDSTA')
CALL CMERR
90 CONTINUE
RETURN
END

```

FILE: DESCEN

ORIGINAL PAGE IS
OF POOR QUALITY

| | | |
|----|--|----------|
| | SUBROUTINE DESCEN(SCN,LNCAT,PTR1,PTR2) | DES00010 |
| | IMPLICIT INTEGER(A-X) | DES00020 |
| | DIMENSION PTR1(LNCAT),PTR2(LNCAT) | DES00030 |
| | REAL SCN(LNCAT),SAVE | DES00040 |
| | J=0 | DES00050 |
| 60 | J=J+1 | DES00060 |
| | IF(J.GT.LNCAT)GO TO 90 | DES00070 |
| | IF(J.EQ.LNCAT)GO TO 75 | DES00080 |
| | IF(SCN(J).LT.SCN(J+1))GO TO 70 | DES00090 |
| | GO TO 60 | DES00100 |
| C | SAVE=SCN(J) | DES00110 |
| 70 | SCN(J)=SCN(J+1) | DES00120 |
| | SCN(J+1)=SAVE | DES00130 |
| C | SAVE1=PTR1(J) | DES00140 |
| | PTR1(J)=PTR1(J+1) | DES00150 |
| | PTR1(J+1)=SAVE1 | DES00160 |
| C | SAVE2=PTR2(J) | DES00170 |
| | PTR2(J)=PTR2(J+1) | DES00180 |
| | PTR2(J+1)=SAVE2 | DES00190 |
| 75 | K=J | DES00200 |
| 80 | IF(K.EQ.1)GO TO 60 | DES00210 |
| | IF(SCN(K).LT.SCN(K-1))GO TO 60 | DES00220 |
| C | SAVE=SCN(K-1) | DES00230 |
| | SCN(K-1)=SCN(K) | DES00240 |
| | SCN(K)=SAVE | DES00250 |
| C | SAVE1=PTR1(K-1) | DES00260 |
| | PTR1(K-1)=PTR1(K) | DES00270 |
| | PTR1(K)=SAVE1 | DES00280 |
| C | SAVE2=PTR2(K-1) | DES00290 |
| | PTR2(K-1)=PTR2(K) | DES00300 |
| | PTR2(K)=SAVE2 | DES00310 |
| | K=K-1 | DES00320 |
| | GO TO 80 | DES00330 |
| 90 | CONTINUE | DES00340 |
| | RETURN | DES00350 |
| | END | DES00360 |
| | | DES00370 |
| | | DES00380 |
| | | DES00390 |
| | | DES00400 |
| | | DES00410 |
| | | DES00420 |

FILE: DSTAPE

```

C*****DST00010
C*
C*      ISOCLS SUBROUTINE
C*
C*      THIS SUBROUTINE GENERATES A CLUSTER IMAGE TAPE IN EITHER
C*      UNIVERSAL OR LARSYS II FORMAT. THE IMAGE MAY OPTIONALLY BE A
C*      ONE CHANNEL TAPE REFLECTING THE CLUSTER NUMBER OF EACH PIXEL.
C*      OR A 'NOFEAT' CHANNEL TAPE REFLECTING THE MEAN VECTOR OF THE
C*      CLUSTER TO WHICH THE PIXEL WAS ASSIGNED.
C*
C*      SUBROUTINE DSTAPE(IPLACE,IBUF,MEANS,/FLDINF/)
C*      IMPLICIT INTEGER(A-X)
C*      INCLUDE COMPK5.LIST
C*      INCLUDE COMPK6.LIST
C*      COMMON/PASS/STOP,LNCAT,NMIN,KRN,STOMAX,OLMIN,SEP,
C*      *      MAP,SPTRIG,IRD,KPTS,NOPTS,PUNCH,
C*      *      ICHN,CHNTHS,ICHAIN(62),NWDS,IBEGIN,REGIN1,
C*      *      REGIN2,BEGIN3,CLSNAM,NOFLD,IPT,TOTWRD,TOTPTS,
C*      *      NCLAS,NOCLS,TOTSUB,TOTFLD,TOTVRT,NOCL,NVRT
C*      *      ,NXTCLS,NOFEAT,MAXCLS,FETVC(30),SYNMTX(62)
C*      *      ,VARSIZ,STATKY,ISOKEY,MAPFMT,MAPKEY,SEQUEN(20),PERCEN,SIMERP
C*      *      ,IORDER,INUNIT,INFILE,INITM,PMIN,SURVEC(62),NOSUB2,CHNVC(30)
C*      *      ,NOCHAN,ERCOMP,NOSEQ,MEANDU,MEANDU,
C*      *      SYMDO,SYMDOU,ITRIGU,ITRIGU,DOFLAG,
C*      *      DUFLAG,DODU,STDOOTS(60),NSDOOTS,SUNCOR(30),LLNLCAT,
C*      *      DVERT(250,2),DRECT(60,2),DVPNT(11,2),IDCNT(2),NDOU(2)
C*      *      ,MXFFT1,MAXPOP
C*      REAL SUNCOR
C*      COMMON/GLOBAL/HEAD(63),MAPTAP,DATAPE,SAVTAP,BMFILE,BMKEY,
C*      *      ,HISFIL,HISKEY,TRFORM,ERIPTP,ERPKEY,MAPUNT,NOFILE,
C*      *      ,DRUMAD,DRMWDS,PAGSIZ,DATEFIL,STAFIL,ASAV,ASAVFL
C*      *      ,NHSTUN,NHSTFI,SCTPUN,MAPFIL
C*      *      ,DOTUNT,DOTFIL,NCHPAS,TRNSFL,BMTRFL,HISTFL,PCHUNT,
C*      *      ,CRDUNT,PRTUNT,RANDIO
CSEND
      DIMENSION IBUF(1),FETVC2(30)
      REAL MEANS(1)
      NC=1
      IF(MAPKEY.EQ.1)NC=NOFEAT
      WRITE(6,HEAD)
C*
      DIMENSION FINE(6),FL(22)
      DIMENSION IPTT(62)
      EQUIVALENCE (FINE(1),LINSTR),(FINE(4),SAMSTR)
      EQUIVALENCE (FINE(2),LINEND),(FINE(5),SAMEND)
      EQUIVALENCE (FINE(3),LININC),(FINE(6),SAMINC)
      DIMENSION IPLACE(NOPTS)
      DIMENSION FLDINF(1)
      DIMENSION FORM(3,2)
      DATA FORM/'UNIV','ERSA','L','LARS','YS I','I'/'
      ICONT=0
      DO 3 I=1,LNCAT
      IPTT(I)=1
3      IF(ICHN.GT.0)GO TO 5
      DO 1 I=1,LNCAT
1      ICHAIN(I)=1
5      CONTINUE
      ADRES=REGIN2
      K=0
      IPFC=IRD
      IPTS=NOPTS
10      IF(IPFC.LE.1)IPTS=KPTS
      IF (IRD.EQ.0) GO TO 20
      CALL PREAD(ADRES,IPLACE,IPTS,ISTAT)
15      IF(ISTAT.EQ.1)GO TO 15
      ADRES=ADRES+IPTS
20      CONTINUE
      IV=5
      DO 50 IFLD=1,NOFLD
      HOREC = 1
      NV=FLDINF(IV+1)
      IR=IV+2*NV+2
      DO 25 I=1,6
25      FINE(I)=FLDINF(IR+I)
      LINES= (LINEND-LINSTR)/LININC +1
      NPTS = (SAMEND-SAMSTR)/SAMINC +1
      NPTS=NC*NPTS
      IF(NPTS.GT.11500)GO TO 80

```

ORIGINAL PAGE IS
OF POOR QUALITY

FILE: DSTAPE

```

LINE = LINSTR-LININC
DO 40 I=1,LINES
C*  ZERO IRUF
C*
DO 26 J=1,NPTS
26 IRUF(J)=0
LINE = LINE + LININC
CALL FDLINT(FLDINF(IV+2),NV,FL,LINE,SAMPS,NFL)
DO 31 L=1,NFL,2
IR=(FL(L)-SAMSTR)/SAMINC+1
IE=(FL(L+1)-SAMSTR)/SAMINC+1
IF(MOD(SAMSTR,SAMINC).NE.MOD(FL(L),SAMINC)) IB=IR+1
IF(IR.GT.IE) GO TO 31
DO 30 J=IB,IE
K=K+1
KP=IPLACE(K)
IF(MAPKEY.EQ.1) GO TO 70
IRUF(J)=ICHAIN(KP)
GO TO 75
70 DO 71 L7=1,NC
JJ=(L7-1)*LPTS+J
KK=(KP-1)*NOFEAT+LZ
71 IRUF(JJ) = MEANS(KK) + 0.5
75 IF(K.LT.IPTS) GO TO 30
IPFC=IREC-1
IF (IREC.EQ.0) GO TO 30
IF (IREC.EQ.1) IPTS=KPTS
CALL WREAD(ADRES,IPLACE,IPTS,ISTAT)
ADRES=ADRES+IPTS
27 IF (ISTAT.EQ.1) GO TO 27
K=0
30 CONTINUE
31 CONTINUE
GO TO (35,37),HDREC
C
C
C
C
35 CONTINUE
LSTLIN = 0
LNFS = 0
IF(MAPKEY.EQ.1) GO TO 72
FETVC2(1)=1
GO TO 73
72 DO 74 KK=1,NOFEAT
74 FETVC2(KK)=KK
73 HDREC=2
NOFILF = NOFILE + 1
CALL WRTHED(NC,FETVC2,LPTS,MAPFMT,MAPUNT)
C
C
C
C
WRITE DATA RECORD
37 LNFS = LNFS + 1
IF(LNFS.EQ.LINES.AND.MAPKEY.NE.1) LSTLIN=-1
CALL WRTLN(IRUF,LSTLIN)
40 CONTINUE
IF(MAPKEY.NE.1) GO TO 48
IF(ORDER.EQ.1) CALL RANK(NOFEAT,FETVC2,LNCAT,MEANS,IPTT)
DO 41 I=1,NPTS
41 IRUF(I)=0
CALL WRTLN(IRUF,LSTLIN)
NTEN=10
JTEN=NTEN
NCLUS=LNCAT
DIV=11
II=1
NRL=LPTS/DIV
IF(NRL.LE.0) NRL=1
IF(LPTS.LT.11) JTEN=LPTS
IF(LPTS.LT.11) DIV=LPTS
51 CNT=NRL
IF((NCLUS-NRL).LT.0) CNT=NCLUS
NCLUS=NCLUS-NRL
IST=NRL*(II-1)+1
IEND=IST+CNT-1
II=II+1
III=0
DO 42 I=IST,IEND
III=III+1
DO 42 K=1,NC

```

DST00800
DST00810
DST00820
DST00830
DST00840
DST00850
DST00860
DST00870
DST00880
DST00890
DST00900
DST00910
DST00920
DST00930
DST00940
DST00950
DST00960
DST00970
DST00980
DST00990
DST01000
DST01010
DST01020
DST01030
DST01040
DST01050
DST01060
DST01070
DST01080
DST01090
DST01100
DST01110
DST01120
DST01130
DST01140
DST01150
DST01160
DST01170
DST01180
DST01190
DST01200
DST01210
DST01220
DST01230
DST01240
DST01250
DST01260
DST01270
DST01280
DST01290
DST01300
DST01310
DST01320
DST01330
DST01340
DST01350
DST01360
DST01370
DST01380
DST01390
DST01400
DST01410
DST01420
DST01430
DST01440
DST01450
DST01460
DST01470
DST01480
DST01490
DST01500
DST01510
DST01520
DST01530
DST01540
DST01550
DST01560
DST01570
DST01580

FILE: DSTAPE

```

      KK=(IPTT(I)-1)*NC*K
      DO 42 J=1,JTEN
      L=(III-1)*DIV*(K-1)*LPTS*J
42  IBUF(L) = MEANS(KK) * 0.5
      DO 43 I=1,NTEN
      ICOUNT=ICOUNT+1
43  CALL WRTLN(IBUF,LSTLIN)
      DO 44 I=1,NPTS
44  IBUF(I)=0
      IF(INCLUS.LE.0) LSTLIN=-1
      CALL WRTLN(IBUF,LSTLIN)
      IF(INCLUS.GT.0) GO TO 51
45  CONTINUE
      WRITE(6,60) NOFILE,FLOINF(IV),(FORM(I,MAPFMT),I=1,3),LNES
      *ICOUNT
60  FORMAT(///T55,'FILE NO.      - ',16,/T55,'FIELD NAME  - ',A4,/
      * T55,'FORMAT      - ',3A4,/T55,'NO. OF SCAN LINES - ',16,/T55,
      *'NO. OF COLOR KEY SCAN LINES - ',16)
      IV = IV + NV*2 + 9
50  CONTINUE
      RETURN
R0  WRITE(4,81)
R1  FORMAT('THE NUMRER OF CHANNELS TIMES THE NUMRER OF SAMPLES HAS
1EXCEEDD 11500.DECREASE THE NUMBER OF CHANNELS OR THE NUMBER OF
2SAMPLES.TERMINATING RUN FROM DSTAPE')
      CALL CMERR
      END

```

DST01590
DST01600
DST01610
DST01620
DST01630
DST01640
DST01650
DST01660
DST01670
DST01680
DST01690
DST01700
DST01710
DST01720
DST01730
DST01740
DST01750
DST01760
DST01770
DST01780
DST01790
DST01800
DST01810
DST01820
DST01830
DST01840
DST01850

FILE: FDLINT

```

SUBROUTINE FDLINT(FIELD,NPTS,FL,YLINE,NSAMP,JJ)
THIS SUBROUTINE WILL RETURN THE PIXEL NUMBERS OF THOSE
PIXELS ON A GIVEN LINE THAT ARE CONTAINED WITHIN THE
BOUNDARIES OF A NON-RECTANGULAR FIELD

INPUT   FIELD - NONE-RECTANGULAR FIELD TABLE
          ALL THE VERTICES MUST BE IN CLOCKWISE
          ORDER AND THE LAST VERTEX HAS TO BE EQUAL
          TO THE FIRST VERTEX FOR FIELD CLOSURE
          THE FIRST VERTEX MUST HAVE MINIMUM
          PIXEL VALUE
          NPTS - NO OF POINTS OF THE N-R FIELD
          YLINE - SCAN LINE NUMBER

OUTPUT  FL - ARRAY CONTAINING THE ORDERED PIXEL INTERCEPTS
          NSAMP - NO OF SAMPLES CONTAINED IN THE FIELD OF
          A GIVEN SCAN LINE
          JJ - THE LENGTH OF THE ARRAY FL

DIMENSION FIELD(2,NPTS),FL(8)
INTEGER X1,X2,Y1,Y2,XX,FL,FIELD,YLINE
INTEGER XNM1,YNM1,XNP2,YNP2
IF(NPTS.EQ.2)GO TO 35
C* ONE VERTEX FIELD
L=YLINE
DO 7 N=1,8
7 FL(N)=0
NPTSE=NPTS-1
I=1
JJ=0
100 X1=FIELD(1,I)
Y1=FIELD(2,I)
J=I+1
X2=FIELD(1,J)
Y2=FIELD(2,J)
IF ( I .EQ. 1 ) GO TO 200
IM1=I-1
XNM1=FIELD(1,IM1)
YNM1=FIELD(2,IM1)
GO TO 300
200 XNM1=FIELD(1,NPTSE)
YNM1=FIELD(2,NPTSE)
300 IP1=I+1
XNP1=FIELD(1,IP1)
YNP1=FIELD(2,IP1)
IF ( I .EQ. NPTSE ) GO TO 400
IP2=I+2
XNP2=FIELD(1,IP2)
YNP2=FIELD(2,IP2)
GO TO 500
400 XNP2=FIELD(1,2)
YNP2=FIELD(2,2)
500 IF ( Y1 .EQ. Y2 ) GO TO 1000
IF ((L.EQ.Y2).AND.(Y2.EQ.YNP2)) GO TO 2000
IF ((L.EQ.Y1).AND.(Y1.EQ.YNM1)) GO TO 2000
RL=L
RX1=X1
RX2=X2
RY1=Y1
RY2=Y2
RXX=((RL-RY1)*(RX2-RX1))/(RY2-RY1)+RX1
XX=RXX+.5
IF(Y1.LT.Y2) GO TO 510
XX=RX1
IF((RXX-XX).GT..5) XX=XX+.1
510 CONTINUE
IF ((XX.GE. X1) .AND. (XX .LE. X2) ) GO TO 600
IF ((XX.LE. X1) .AND. (XX .GE. X2) ) GO TO 600
2000 I=I+1
IF ( I .GT. NPTSE ) GO TO 5
GO TO 100
600 IF(L.LE.Y1.AND.L.GE.Y2) GO TO 700
IF(L.LE.Y2.AND.L.GE.Y1) GO TO 700
GO TO 2000
700 JJ=JJ+1
FL(JJ)=XX
IF ( JJ .EQ. 1 ) GO TO 2000

```

ORIGINAL PAGE IS
OF POOR QUALITY

FDL00010
FDL00020
FDL00030
FDL00040
FDL00050
FDL00060
FDL00070
FDL00080
FDL00090
FDL00100
FDL00110
FDL00120
FDL00130
FDL00140
FDL00150
FDL00160
FDL00170
FDL00180
FDL00190
FDL00200
FDL00210
FDL00220
FDL00230
FDL00240
FDL00250
FDL00260
FDL00270
FDL00280
FDL00290
FDL00300
FDL00310
FDL00320
FDL00330
FDL00340
FDL00350
FDL00360
FDL00370
FDL00380
FDL00390
FDL00400
FDL00410
FDL00420
FDL00430
FDL00440
FDL00450
FDL00460
FDL00470
FDL00480
FDL00490
FDL00500
FDL00510
FDL00520
FDL00530
FDL00540
FDL00550
FDL00560
FDL00570
FDL00580
FDL00590
FDL00600
FDL00610
FDL00620
FDL00630
FDL00640
FDL00650
FDL00660
FDL00670
FDL00680
FDL00690
FDL00700
FDL00710
FDL00720
FDL00730
FDL00740
FDL00750
FDL00760
FDL00770
FDL00780
FDL00790

FILE: FDLINT

```

      IF ( I.NE. NPTSE ) GO TO 3000
      IF ( L.NE. Y2 ) GO TO 3000
      XNM1=X1
      YNM1=Y1
      X1=X2
      Y1=Y2
      X2=FIELD(1,2)
      Y2=FIELD(2,2)
      GO TO 3001
3000  IF ( L.NE. Y1 ) GO TO 2000
3001  IF ((Y1.LT. YNM1) .AND. (Y1.GT. Y2 )) GO TO 4000
      IF ((Y1.GT. YNM1) .AND. (Y1.LT. Y2 )) GO TO 4000
      GO TO 2000
4000  FL(JJ) = 0
      JJ = JJ+1
      GO TO 2000
1000  IF ( L.NE. Y1 ) GO TO 2000
      IF (X1.GT. X2) GO TO 5000
      IF (YNM1.LT. Y1) GO TO 6000
      IF ( YNP2.GT. Y2 ) GO TO 7000
      JJ = JJ+1
      FL(JJ) = X1
      GO TO 2000
7000  JJ = JJ+1
      FL(JJ) = X1
      MM = JJ+1
      FL(MM) = X2
      JJ = MM
      GO TO 2000
6000  IF ( YNP2.LT. Y2 ) GO TO 2000
      JJ = JJ+1
      FL(JJ) = X2
      GO TO 2000
5000  IF ( YNM1.LT. Y1 ) GO TO 9000
      IF ( YNP2.GT. Y2 ) GO TO 2000
      JJ = JJ+1
      FL(JJ) = X2
      IF (NPTSE.EQ.2) FL(JJ)=X1
      GO TO 2000
9000  IF ( YNP2.GT. Y2 ) GO TO 8000
      JJ = JJ+1
      FL(JJ) = X1
      MM = JJ+1
      FL(MM) = X2
      JJ = MM
      GO TO 2000
8000  JJ = JJ+1
      FL(JJ) = X1
      GO TO 2000
5  NPTS1 = JJ-1
   DO 29 NI = 1,NPTS1
     NP1 = NI+1
     DO 29 NJ = NP1,JJ
       IF ( FL(NI) - FL(NJ) ) 29,29,28
28  NTEMP = FL(NI)
     FL(NI) = FL(NJ)
     FL(NJ) = NTEMP
29  CONTINUE
   NSAMP = 0
   DO 30 N = 1,JJ,2
     NN = N+1
     NSAMP = NSAMP+(FL(NN) -FL(N)+1)
30  CONTINUE
   RETURN
35  IF (YLINE.NE.FIELD(2,1)) RETURN
     FL(1)=FIELD(1,1)
     FL(2)=FIELD(1,1)
     NSAMP=1
     JJ=2
     RETURN
   END

```

FDL00800
 FDL00810
 FDL00820
 FDL00830
 FDL00840
 FDL00850
 FDL00860
 FDL00870
 FDL00880
 FDL00890
 FDL00900
 FDL00910
 FDL00920
 FDL00930
 FDL00940
 FDL00950
 FDL00960
 FDL00970
 FDL00980
 FDL00990
 FDL01000
 FDL01010
 FDL01020
 FDL01030
 FDL01040
 FDL01050
 FDL01060
 FDL01070
 FDL01080
 FDL01090
 FDL01100
 FDL01110
 FDL01120
 FDL01130
 FDL01140
 FDL01150
 FDL01160
 FDL01170
 FDL01180
 FDL01190
 FDL01200
 FDL01210
 FDL01220
 FDL01230
 FDL01240
 FDL01250
 FDL01260
 FDL01270
 FDL01280
 FDL01290
 FDL01300
 FDL01310
 FDL01320
 FDL01330
 FDL01340
 FDL01350
 FDL01360
 FDL01370
 FDL01380
 FDL01390
 FDL01400
 FDL01410
 FDL01420
 FDL01430
 FDL01440
 FDL01450
 FDL01460
 FDL01470
 FDL01480
 FDL01490
 FDL01500

FILE: FIND12

| | | | |
|-------|--|---|--------|
| C | FUNCTION FIND12(CARD,COL,VECTOR) | F | N00010 |
| | IMPLICIT INTEGER (A-H,O-Z) | F | N00020 |
| | ----- | F | N00030 |
| | | F | N00040 |
| | | F | N00050 |
| | | F | N00060 |
| | | F | N00070 |
| | CALL.. J = FIND12(CARD,COL,VECTOR) | F | N00080 |
| | ARGS.. CARD = BCD BUFFER | F | N00090 |
| | COL = PTR TO POSTION IN 'CARD' | F | N00100 |
| | VECTOR = VEC CONTAINING N SYMBOLS | F | N00110 |
| | TO BE LOCATED IN CARD | F | N00120 |
| | (N IS GIVEN IN VECTOR(1)) | F | N00130 |
| | EG. /2.'S'. '1'/' | F | N00140 |
| | | F | N00150 |
| | | F | N00160 |
| | REQUIRES. NOR | F | N00170 |
| | PURPOSE.. USED TO LOCATE SPECIAL SYMBOLS IN 'CARD' | F | N00180 |
| | | F | N00190 |
| | RETURNS.. COL = PTS AT SYMROL (IF LOCATED) | F | N00200 |
| | J = PTS AT SYMBOL LOCATED IN 'VECTOR' | F | N00210 |
| | ----- | F | N00220 |
| | | F | N00230 |
| | | F | N00240 |
| | | F | N00250 |
| | | F | N00260 |
| | | F | N00270 |
| | | F | N00280 |
| | | F | N00290 |
| | | F | N00300 |
| | | F | N00310 |
| | | F | N00320 |
| | | F | N00330 |
| | | F | N00340 |
| | | F | N00350 |
| | | F | N00360 |
| | | F | N00370 |
| | | F | N00380 |
| | | F | N00390 |
| | | F | N00400 |
| | | F | N00410 |
| | | F | N00420 |
| | | F | N00430 |
| | | F | N00440 |
| | | F | N00450 |
| | | F | N00460 |
| | | F | N00470 |
| | | F | N00480 |
| | | | |
| C | DIMENSION CARD(1),VECTOR(1) | | |
| | DATA CRDSIZ/62/ | | |
| | K = VECTOR(1)+1 | | |
| | L = COL+1 | | |
| | DO 10 COL=L,CRDSIZ | | |
| | DO 10 I=2,K | | |
| | II = I | | |
| | IF (CARD(COL).EQ.VECTOR(I)) GO TO 20 | | |
| 10 | CONTINUE | | |
| | I = -1 | | |
| | COL = L-1 | | |
| 20 | FIND12 = I | | |
| C | WRITE(6,102) (CARD(K),K=1,62),COL,I,VECTOR(I) | | |
| C 102 | FORMAT(' FIND12 ENTERED'/ ' ',62A1,I10/' ',15,A4) | | |
| | RETURN | | |
| | END | | |

ORIGINAL PAGE IS
OF POOR QUALITY

FILE FLOINT

```

SUBROUTINE FLOINT (/BLOCK/, /FETVEC/, /NOFEAT) -
IMPLICIT INTEGER (A-Z)
C*
C* ENTRY FOR POSITIONING TAPE TO CORRECT SCAN LINE FOR A SPECIFIC FILE
C*
COMMON /TAPERD/ IUNIT, IFIRST, FSCAN, SAMEND, SAMINC, READY, NSCAN,
* LINC, IU(200), DSL, LHUF(30), JREC(30), IAYTE(30), NRUFS, FILENO, LINEN
* LININC, NSAMP, NOCHAN, FOMMT
DIMENSION BLOCK(6)
DIMENSION FETVEC(NOFEAT)
EQUIVALENCE ((ID(1), NRPOS), (ID(2), NCPR),
* ((ID(3), NPWC), (ID(4), ANCLNG),
* ((ID(5), NC), (ID(6), NS),
* ((ID(7), NRITS), (ID(8), DOI),
* ((ID(9), NDSPR), (ID(10), NCAR),
* ((ID(11), SVD), (ID(16), PRSZ)
READY = 1
NOCHAN = NOFEAT
C* CHECK FETVEC
DO 125 I=1, NOFEAT
IF (FETVEC(I) .GT. NC) GO TO 126
125 CONTINUE
GO TO 127
124 WRITE(6,470) NC
CALL CMERR
127 CONTINUE
LINSTR=BLOCK(1)
IF (LINSTR.GE. IFIRST) GO TO 130
WRITE(6,430) IFIRST
CALL CMERR
130 CONTINUE
IF (BLOCK(2).GE. IFIRST) GO TO 132
WRITE(6,430) IFIRST
CALL CMERR
132 CONTINUE
IF (FOMMT.EQ.3) GO TO 200
IF (FOMMT.EQ.4) GO TO 220
C* FLINE=FIRST SCAN ON RECORD CONTAINING LINSTR
FLINE=LINSTR-MOD((LINSTR-IFIRST), NDSPR)
LSKIP=((FLINE-FSCAN)/NDSPR-1)*NRPOS
IF (LSKIP) 135, 134, 138
135 FSKIP = ((BLOCK(1) - IFIRST) / NDSPR) * NRPOS + 1
C*
C* ON MULTI-FILE TAPES FOR FILES OTHER THAN FILE 1, DO THE FOLLOWING-
C* 1. BACK SPACE 1 FILE
C* 2. HEAD FORWARD 1 E-O-F
C* 3. HEAD FORWARD NO. OF DESIRED RECORDS
C*
IF (FILENO.EQ. 0 .AND. FSKIP .LE. IABS(LSKIP)) GO TO 136
FSKIP = IABS(LSKIP)
DO 134 II=1, FSKIP
134 BACKSPACE IUNIT
GO TO 139
C*
C* FOR FILE 1 DO A REWIND AND SKIP FORWARD THE DESIRED NO. OF RECORDS
C*
136 REWIND IUNIT
DO 137 II=1, FSKIP
137 READ(IUNIT,480) DUMMY
GO TO 139
C*
C* SKIP DOWN THE TAPE TO BEGINNING LINE OF THIS FIELD.
C*
138 IF (FSCAN.EQ.FLINE) GO TO 140
IF (LSKIP .EQ. 0) GO TO 139
DO 141 II=1, LSKIP
141 READ(IUNIT,480) DUMMY
139 CONTINUE
FSCAN=FLINE
140 CONTINUE
NSCAN=LINSTR
IF (BLOCK(5).LE. NS) GO TO 145
WRITE(6,440) NS
CALL CMERR
145 IF (BLOCK(4).LE. NS) GO TO 146
WRITE(6,440) NS

```

FLD000010
 FLD000020
 FLD000030
 FLD000040
 FLD000050
 FLD000060
 FLD000070
 FLD000080
 FLD000090
 FLD000100
 FLD000110
 FLD000120
 FLD000130
 FLD000140
 FLD000150
 FLD000160
 FLD000170
 FLD000180
 FLD000190
 FLD000200
 FLD000210
 FLD000220
 FLD000230
 FLD000240
 FLD000250
 FLD000260
 FLD000270
 FLD000280
 FLD000290
 FLD000300
 FLD000310
 FLD000320
 FLD000330
 FLD000340
 FLD000350
 FLD000360
 FLD000370
 FLD000380
 FLD000390
 FLD000400
 FLD000410
 FLD000420
 FLD000430
 FLD000440
 FLD000450
 FLD000460
 FLD000470
 FLD000480
 FLD000490
 FLD000500
 FLD000510
 FLD000520
 FLD000530
 FLD000540
 FLD000550
 FLD000560
 FLD000570
 FLD000580
 FLD000590
 FLD000600
 FLD000610
 FLD000620
 FLD000630
 FLD000640
 FLD000650
 FLD000660
 FLD000670
 FLD000680
 FLD000690
 FLD000700
 FLD000710
 FLD000720
 FLD000730
 FLD000740
 FLD000750
 FLD000760

FILE FLDINT

```

C      CALL CMERR
C      SKIP FORWARD NECESSARY RECORDS FOR LANDSAT 1 OR 2
200    DO 210 I=1,LINSTR
      READ(IUNIT,480)DUMMY
210    CONTINUE
      GO TO 146
C      SKIP RECORDS FOR LANDSAT III
220    SKIP=NRPDS*(LINSTR-1)
      IF (SKIP.EQ.0) GO TO 146
      DO 230 I=1,SKIP
        READ(IUNIT,480)DUMMY
230    CONTINUE
146  CONTINUE
      LINEND=BLOCK(2)
      LININC=BLOCK(3)
      SAMSTR=BLOCK(4)
      SAMEND=BLOCK(5)
      SAMINC=BLOCK(6)
C      LINC=NO. OF RECORDS TO SKIP AFTER EACH SCAN LINE
      LINC=(LININC/NRPPS - 1)*NRPPS
      IF (LINC.LT.0) LINC=0
C      ESTABLISH AREAS ON EACH SCAN LINE TO UNPACK
      IF (FORMAT.EQ.3) GO TO 1000
      IF (FORMAT.EQ.4) GO TO 2000
      ANC=ANCLNG + SAMSTR + SVD - 1
      IF (FORMAT.EQ.1) ANC = ANC + 2
      NBUFS=NRPPS/10
      IF (MOD(NRPPS,10).NE.0) NBUFS=NBUFS+1
      FC=1
      LC=NCAR
      K=1
      DO 190 I=1,NOFEAT
        TRY = 0
185    CONTINUE
        DO 170 JREC=K,NRPPS
          IF (JREC.GT.1) ANC=2 + SAMSTR + SVD - 1
          IF (FETVEC(I).GE.FC.AND.FETVEC(I).LE.LC) GO TO 150
          IF (FETVEC(I).GT.LC.AND.JREC.LT.NRPPS) GO TO 160
          FC = 1
          LC = NCAR
          K = 1
          ANC = (ANCLNG + 2) + SAMSTR + SVD - 1
          TRY = TRY + 1
          IF (TRY.LE.2) GO TO 185
          WRITE (A,380)FETVEC(I)
          CALL CMERR
150    JBYTE(I)=(FETVEC(I)-FC)*NS + ANC
          JREC(I)=MOD(JREC,10)
          IF (JREC(I).EQ.0) JREC(I)=10
          LBUF(I)=JREC/10 + 1
          IF (MOD(JREC,10).EQ.0) LBUF(I)=LBUF(I)-1
          GO TO 140
160    FC=LC+1
          LC=LC+NCPR
170    CONTINUE
180    K=JREC
190    CONTINUE
C      NSAMP = NO. OF SAMPLES TO UNPACK FOR EACH FEATURE IN FETVEC
C      NSAMP = (SAMEND - SAMSTR) / SAMINC + 1
900    RETURN
C      SET UP IRYTE FOR LANDSAT 1 OR 2
1000   JREC(I)=SAMSTR
        NSCAN=LINSTR
        FSCAN=LINSTR
        DO 1100 I=1,NOFEAT

```

FLD00770
 FLD00780
 FLD00790
 FLD00800
 FLD00810
 FLD00820
 FLD00830
 FLD00840
 FLD00850
 FLD00860
 FLD00870
 FLD00880
 FLD00890
 FLD00900
 FLD00910
 FLD00920
 FLD00930
 FLD00940
 FLD00950
 FLD00960
 FLD00970
 FLD00980
 FLD00990
 FLD01000
 FLD01010
 FLD01020
 FLD01030
 FLD01040
 FLD01050
 FLD01060
 FLD01070
 FLD01080
 FLD01090
 FLD01100
 FLD01110
 FLD01120
 FLD01130
 FLD01140
 FLD01150
 FLD01160
 FLD01170
 FLD01180
 FLD01190
 FLD01200
 FLD01210
 FLD01220
 FLD01230
 FLD01240
 FLD01250
 FLD01260
 FLD01270
 FLD01280
 FLD01290
 FLD01300
 FLD01310
 FLD01320
 FLD01330
 FLD01340
 FLD01350
 FLD01360
 FLD01370
 FLD01380
 FLD01390
 FLD01400
 FLD01410
 FLD01420
 FLD01430
 FLD01440
 FLD01450
 FLD01460
 FLD01470
 FLD01480
 FLD01490
 FLD01500
 FLD01510
 FLD01520

FILE FLDINT

| | | |
|-------------------------|---|-----------|
| 1100 | IHYTE(I)=1+(FETVEC(I)-1)*2 | FLD001530 |
| | CONTINUE | FLD001540 |
| | GO TO 900 | FLD001550 |
| C | | FLD001560 |
| C | SET UP IHYTE AND NSCAN FOR LANDSAT III | FLD001570 |
| C | IHYTE IS FETVEC FOR LANDSAT III | FLD001580 |
| 2000 | DO 2100 I=1,NOFEAT | FLD001590 |
| | IHYTE(I)=FETVEC(I) | FLD001600 |
| 2100 | CONTINUE | FLD001610 |
| | NSCAN=LINSTR | FLD001620 |
| | FSCAN=LINSTR | FLD001630 |
| | LHUF(I)=SAMSTR | FLD001640 |
| | GO TO 900 | FLD001650 |
| 380 | FORMAT(' FEATURE NUMBERS',15,' AND ABOVE ARE NOT ON DATA TAPE', | FLD001660 |
| •) | | FLD001670 |
| 430 | FORMAT(' FIRST SCAN ON THIS TAPE IS NUMBERED',16,' FIELD DEFINITION | FLD001680 |
| •N IN ERROR') | | FLD001690 |
| 440 | FORMAT(' NUMBER OF SAMPLES OF PER SCAN ON THIS TAPE IS',16,' FIELD | FLD001700 |
| • DEFINITION IN ERROR') | | FLD001710 |
| 470 | FORMAT(' THIS TAPE CONTAINS ONLY',16,' CHANNELS') | FLD001720 |
| 480 | FORMAT(1A4) | FLD001730 |
| | END | FLD001740 |
| | | FLD001750 |

FILE: FLTNUM

```
C      FUNCTION FLTNUM(CARD,COL,NUMVEC,VECMAX)
      IMPLICIT INTEGER (A-H,O-Z)
-----
      CALL...    J = FLTNUM(CARD,COL,NUMVEC,VECMAX)
      ARGS...    CARD   - 62 COL CARD BUFFER
                  COL    - PTR TO FIRST COL IN CARD TO SCAN
                  NUMVEC - BUFFER IN WHICH TO RETURN THE NUMBERS
                  VECSIZ - LENGTH OF NUMVEC
      REQUIRES.. NONE
      PURPOSE..  INTERPRETS REAL NUMBERS SEPARATED BY COMMAS ON CARD
                  AND RETURNS THEM IN NUMVEC.
                  STOPS AT FIRST 'NONNUMERIC'
                  (NOTE. NUMBERS MAY APPEAR IN
                    'DATA STATEMENT FORMAT' )
      RETURNS..  COL    - COLUMN WHERE SCAN TERMINATED
                  NUMVEC - VECTOR OF REAL NUMBERS FOUND
                  FLTNUM - NO OF REAL NUMBERS RETURNED
-----
      REAL NUMVEC(20),PNUM
      DIMENSION CARD(62)
      DATA BLANK/' ',COMMA/',' ,PLUS/'+' ,MINUS/'-' ,STAR/'*',
1  PERIOD/'.' ,ZERO/'0' ,CRDSIZ/62/
      LOGICAL*1 LLNUM(1)
      LOGICAL*1 LNUM(4)
      DIMENSION INUM(1)
      EQUIVALENCE (INUM(1),LNUM(1))
      DATA XX/Z000000F0/
      DATA LLNUM/Z00/
-----
      L = COL+1
      VECPOS = 1
10  WNUM = 0
      PCNT = 0
      PNUM = 0.0
      ITR = 1
      SIDE = -1
      SIGN = +1
C
      DO 60 COL=L,CRDSIZ
      IF (CARD(COL).EQ.BLANK) GO TO 60
      IF (CARD(COL).EQ.PLUS) GO TO 60
      IF (CARD(COL).EQ.COMMA) GO TO 70
      IF (CARD(COL).NE.MINUS) GO TO 20
      SIGN = -SIGN
      GO TO 60
20  IF (CARD(COL).NE.STAR) GO TO 30
      ITR = WNUM
      WNUM = 0
      PCNT = 0
      PNUM = 0.0
      SIDE = -1
      SIGN = +1
      GO TO 60
30  IF (CARD(COL).NE.PERIOD) GO TO 40
      SIDE = 1
      GO TO 60
40  IF (CARD(COL).LT.ZERO) GO TO 90
      INUM(1)=CARD(COL)
      LNUM(4)=LNUM(1)
      LNUM(1)=LLNUM(1)
      LNUM(2)=LLNUM(1)
      LNUM(3)=LLNUM(1)
      MORNUM=INUM(1)-XX
      IF (SIDE.LT.0) GO TO 50
      PCNT = PCNT+1
      PNUM = PNUM+MORNUM*(0.1**PCNT)
```

ORIGINAL PAGE IS
OF POOR QUALITY

FILE: FLTNUM

| | |
|---|----------|
| GO TO 60 | FLT00800 |
| 50 WNUM = 10*WNUM+MORNUM | FLT00810 |
| 60 CONTINUE | FLT00820 |
| C COL = CRDSIZ+1 | FLT00830 |
| GO TO 90 | FLT00840 |
| 70 VECFIN = VECPOS+ITER-1 | FLT00850 |
| IF (VECFIN.GT. VECMAX) VECFIN = VECMAX-1 | FLT00860 |
| DO 80 I=VECPOS,VECFIN | FLT00870 |
| 80 NUMVEC(I) = SIGN*(WNUM+PNUM) | FLT00880 |
| L = COL+1 | FLT00890 |
| VECPOS = VECFIN+1 | FLT00900 |
| IF (VECPOS.LE.VECMAX) GO TO 10 | FLT00910 |
| GO TO 110 | FLT00920 |
| 90 COL = COL-1 | FLT00930 |
| VECFIN = VECPOS+ITER-1 | FLT00940 |
| IF (VECFIN.GT. VECMAX) VECFIN = VECMAX | FLT00950 |
| DO 100 I=VECPOS,VECFIN | FLT00960 |
| 100 NUMVEC(I) = SIGN*(WNUM+PNUM) | FLT00970 |
| 110 FLTNUM = VECFIN | FLT00980 |
| C WRITE(6,706) (CARD(K),K=1,62), COL,FLTNUM,(NUMVEC(K),K=1,FLTNUM) | FLT00990 |
| C 706 FORMAT(' FLTNUM ENTERED'/, ' ', 62A1,110/, ' ', 15,20F8.2,/, ' ', 10F8.2) | FLT01000 |
| RETURN | FLT01010 |
| END | FLT01020 |
| | FLT01030 |

FILE: FSRFSL

```
SUBROUTINE FSRFSL(UNIT,FILE,ISTAT)
IMPLICIT INTEGER (A-Z)
N=0
C
  ISTAT = 0
  IF (FILE .EQ. 0) RETURN
  IF (FILE .LT. 0) GO TO 100
C
  MOVE UNIT FORWARD N E-O-F'S
C
40  READ(UNIT,END=50)
   GO TO 40
50  N = N + 1
   IF (N .EQ. FILE) RETURN
   GO TO 40
C
100 WRITE(6,110)
110 FORMAT(' FSRFSL ONLY SKIPS FORWARD*')
   ISTAT = 2
   RETURN
C
  END
```

FSR00010
FSR00020
FSR00030
FSR00040
FSR00050
FSR00060
FSR00070
FSR00080
FSR00090
FSR00100
FSR00110
FSR00120
FSR00130
FSR00140
FSR00150
FSR00160
FSR00170
FSR00180
FSR00190
FSR00200
FSR00210
FSR00220

ORIGINAL PAGE IS
OF POOR QUALITY

FILE: FFSMFL

```
SUBROUTINE FFSMFL(UNIT,FILE,ISTAT)
IMPLICIT INTEGER (A-Z)
N=0
C
  ISTAT = 0
  IF (FILE.EQ. 0) RETURN
  IF (FILE.LT. 0) GO TO 100
C
  MOVE UNIT FORWARD N E-O-F'S
C
  40 READ(UNIT,30,END=50) DUMMY
  30 FORMAT(1A4)
  GO TO 40
  50 N = N + 1
  IF (N.EQ. FILE) RETURN
  GO TO 40
C
  100 WRITE(6,110)
  110 FORMAT(' FFSMFL ONLY SKIPS FORWARD!')
  ISTAT = 2
  RETURN
C
  END
```

FSB00010
FSB00020
FSB00030
FSB00040
FSB00050
FSB00060
FSB00070
FSB00080
FSB00090
FSB00100
FSB00110
FSB00120
FSB00130
FSB00140
FSB00150
FSB00160
FSB00170
FSB00180
FSB00190
FSB00200
FSB00210
FSB00220
FSB00230

FILE: GETINF

```
      SURROUTINE GETINF (ARRAY, FLDSAV, VERTEX, CLSNMS, NOSUBS, SUBNM, NOCLS,  
      * TOTSUB)  
C      IMPLICIT INTEGER (A-Z)  
C      DIMENSION ARRAY(1), FLDSAV(4,1), VERTEX(1), CLSNMS(1), NOSUBS(1),  
      * SUBNM(1), CLSVEC(60)  
C      JJ = 0  
C      KPT = 1  
C      NFS = 0  
C      L = 0  
C      SEE SURROUTINE RDATA FOR STORAGE ARRANGEMENT OF 'ARRAY'  
C      DO 80 CLS=1, NOCLS  
C      L = L + NFS  
C      CLSNMS(CLS) = ARRAY(KPT)  
C      NOSUBS(CLS) = ARRAY(KPT+2)  
C      NFS = ARRAY(KPT+3)  
C      IKP = KPT + 4  
C      DO 100 I=1, NFS  
C      FLDSAV(1, I+L) = ARRAY(IKP)  
C      FLDSAV(2, I+L) = CLS  
C      FLDSAV(3, I+L) = 0  
C      FLDSAV(4, I+L) = ARRAY(IKP+1)  
C      NV = FLDSAV(4, I+L)*2  
C      DO 90 J=1, NV  
C      90 VERTEX(JJ+J) = ARRAY(IKP+1+J)  
C      JJ = NV + JJ  
C      100 CONTINUE  
C      IKP = IKP + NV + 9  
C      KPT = ARRAY(KPT+1)  
C      80 CONTINUE  
C      K = 0  
C      DO 120 I=1, NOCLS  
C      NSUB = NOSUBS(I)  
C      DO 120 J=1, NSUB  
C      K = K + 1  
C      120 CLSVEC(K) = I  
C      CALL NAMSTA (SUBNM, CLSVEC, NOSUBS, TOTSUB, CLSNMS, NOCLS)  
C      RETURN  
C      END
```

FILE: GETST

```

SUBROUTINE GETST(UNIT,FILE,MENS,STDEV,NOSUB2,SUBVEC,NOCHAN,
* CHNVEC,MEANS,COVAR,ITRIG)
*****
SUBROUTINE GETSTA RETRIEVES THE MEANS AND STANDARD DEVIATIONS
FROM A STATISTICS FILE IN THE LARSYS 'SAVTAP' FORMAT.
INPUT ARGUMENTS:
UNIT - FORTRAN UNIT NUMBER FROM WHICH THE STATS ARE TO BE
      RETRIEVED.
FILE - FILE NO. ON 'UNIT' FROM WHICH THE STATS ARE TO BE
      RETRIEVED.
NOCHAN - NO. OF CHANNELS REQUESTED FROM TRAINING SEGMENT.
* CHNVEC - NOCHAN LESS THAN OR = NO. OF CHANNELS ON STAT FILE
          ARRAY CONTAINING ACTUAL CHANNELS REQUESTED FROM
          TRAINING SEGMENT. MUST BE A SUBSET OF CHANNELS
          ON STAT FILE
ITRIG - IF ITRIG=1, ST. DEV. WILL BE RETURNED ALONG WITH MEANS.
OUTPUT ARGUMENTS:
MENS - ARRAY CONTAINING THE MEAN VECTORS FOR EACH SUBCLASS
      (A SUBSET OF THE CHANNELS MAY BE SELECTED, BUT NOT
      A SUBSET OF THE SUBCLASSES).
STDEV - ARRAY CONTAINING THE SUBSET OF ST. DEV. FOR REQUES-
        TED CHANNELS IN EACH SUBCLASS
        IN EACH SUBCLASS.
NOSUB - NUMBER OF SUBCLASSES ON THE STAT FILE
CONTINUE
CHNVEC - ARRAY CONTAINING ACTUAL CHANNELS REQUESTED FROM
        TRAINING SEGMENT
*****
**NOTE** THE STORAGE ARRAYS PASSED TO THIS SUBROUTINE FOR THE
MEANS AND STANDARD DEVIATIONS SHOULD BE SINGLY DIMENSIONED
IN THE CALLING ROUTINE. ON OUTPUT THE ITEMS ARE STORED
AS FOLLOWS: (SAME FOR STDEV)
      MEANS(1) - CHANNEL 1, SUBCLASS 1
      (2) - CHANNEL 2, SUBCLASS 1
      3 - CHANNEL 3, SUBCLASS 1
      .
      (NOCHAN) - CHANNEL NOCHAN, SUBCLASS 1
      (NOCHAN+1) - CHANNEL 1, SUBCLASS 2
      (NOCHAN+2) - CHANNEL 2, SUBCLASS 2
      (NOCHAN+3) - CHANNEL 3, SUBCLASS 2
      .
      (2*NOCHAN) - CHANNEL NOCHAN OF SUBCLASS 2
      .
      ETC.
      THRU
      (NOCHAN*NOSUB)
CONTINUE
*****
IMPLICIT INTEGER (A-Z)
DIMENSION CHNVEC(30),DUMVEC(30),CHNVEC(1)
DATA BLANK/' '
DIMENSION SUBVEC(1)
DIMENSION MEANS(1),STDEV(1)
REAL MEANS,STDEV,COVAR(465)
REAL MENS(1)
REWIND UNIT
NF=FILE-1
CALL FSAFSL(UNIT,NF,ISTAT1)
IF(ISTAT1.EQ.0)GO TO 5
WRITE(6,100)UNIT,FILE
100 FORMAT(1X,'ERROR IN POSITIONING UNIT',I3,' TO FILE',I3)
CALL CMERR

```

GET00010
 GET00020
 GET00030
 GET00040
 GET00050
 GET00060
 GET00070
 GET00080
 GET00090
 GET00100
 GET00110
 GET00120
 GET00130
 GET00140
 GET00150
 GET00160
 GET00170
 GET00180
 GET00190
 GET00200
 GET00210
 GET00220
 GET00230
 GET00240
 GET00250
 GET00260
 GET00270
 GET00280
 GET00290
 GET00300
 GET00310
 GET00320
 GET00330
 GET00340
 GET00350
 GET00360
 GET00370
 GET00380
 GET00390
 GET00400
 GET00410
 GET00420
 GET00430
 GET00440
 GET00450
 GET00460
 GET00470
 GET00480
 GET00490
 GET00500
 GET00510
 GET00520
 GET00530
 GET00540
 GET00550
 GET00560
 GET00570
 GET00580
 GET00590
 GET00600
 GET00610
 GET00620
 GET00630
 GET00640
 GET00650
 GET00660
 GET00670
 GET00680
 GET00690
 GET00700
 GET00710
 GET00720
 GET00730
 GET00740
 GET00750
 GET00760
 GET00770
 GET00780
 GET00790

FILE: GETST

```

5 CONTINUE
  READ(UNIT) NOCLS, NOSUB, NCHAN, NOFLD, TOTVRT, (CHNVC1(I), I=1, NCHAN)
C*
C*   DEFAULT ALL SUBCLASSES FROM STATISTICS FILE
C*
  IF (NOSUB2.NE.0) GO TO 7
  DO 6 I=1, NOSUB
    SURVEC(I)=I
  6   NOSUB2=NOSUB
  7   CONTINUE
  DO 77 I=1, NOSUB2
    IF (SURVEC(I).LE. NOSUB) GO TO 77
    WRITE(6,70) NOSUB
  70  FORMAT(/' REQUESTED SUBCLASS IS NOT ON STAT FILE. STAT FILE CONTAINS
    *NS ', I3, ' SUBCLASSES')
    CALL CMERR
  77  CONTINUE
C*
C*   DEFAULT -- ALL CHANNELS FROM STAT FILE
C*
  IF (NOCHAN.NE.0) GO TO 9
  DO 8 I=1, NCHAN
    CHNVEC(I) = CHNVC1(I)
  8   NOCHAN = NCHAN
  9   CONTINUE
C*
C*   READ PAST THE TRAINING FIELD INFORMATION
C*
  DO 10 I=1, NOFLD
    READ(UNIT) DUM
    READ(UNIT) DUM
  10  CONTINUE
  READ(UNIT) DUM
  VARSIZ = NCHAN*(NCHAN+1)/2
  MB=1
  ME = NCHAN
  IC=1
  DO 30 J=1, NOSUB
    READ(UNIT) N, (COVAR(J), J=1, VARSIZ), (MEANS(J), J=MB, ME)
    IF (SURVEC(IC).NE.I) GO TO 30
    IF (ITRIG.NE.1) GO TO 25
C*
C*   GET STANDARD DEVIATIONS
C*
    JK=0
    DO 20 JA=1, NOCHAN
      JK= JK + JA
      STDEV(MB+JA-1) = SQRT(COVAR(JK))
    20 CONTINUE
    25 CONTINUE
    IC = IC + 1
    MB = MB + NCHAN
    ME = ME + NCHAN
    30 CONTINUE
C*
C*   GET SURSET OF MEANS
C*   AND GET SUBSET OF ST. DEV.
C*
    DO 50 J=1, NOCHAN
      DO 40 K=1, NCHAN
        IF (CHNVEC(J).EQ. CHNVC1(K)) GO TO 50
    40 CONTINUE
    WRITE(6,110) CHNVEC(J), (CHNVC1(L), L=1, NCHAN)
  110 FORMAT(' CHANNEL NO. ', I2, ' IS NOT ON TRAINING STAT FILE. CHANNELS
    1 ARE / 10X.30(I2,1X))
    CALL EXIT
    50 DUMVEC(J) = K
C
    JJJ = 0
    DO 60 K=1, NOSUB2
      DO 60 J=1, NOCHAN
        JJ = DUMVEC(J) + (K-1)*NCHAN
        JJJ = JJJ + 1
      MEANS(JJJ) = MEANS(JJ)
    60 IF (ITRIG.NE.0) STDEV(JJJ) = STDEV(JJ)
    200 FORMAT(/'157, 'MEANS')
    ISTART = 1
    IEND = 12

```

GET00800
 GET00810
 GET00820
 GET00830
 GET00840
 GET00850
 GET00860
 GET00870
 GET00880
 GET00890
 GET00900
 GET00910
 GET00920
 GET00930
 GET00940
 GET00950
 GET00960
 GET00970
 GET00980
 GET00990
 GET01000
 GET01010
 GET01020
 GET01030
 GET01040
 GET01050
 GET01060
 GET01070
 GET01080
 GET01090
 GET01100
 GET01110
 GET01120
 GET01130
 GET01140
 GET01150
 GET01160
 GET01170
 GET01180
 GET01190
 GET01200
 GET01210
 GET01220
 GET01230
 GET01240
 GET01250
 GET01260
 GET01270
 GET01280
 GET01290
 GET01300
 GET01310
 GET01320
 GET01330
 GET01340
 GET01350
 GET01360
 GET01370
 GET01380
 GET01390
 GET01400
 GET01410
 GET01420
 GET01430
 GET01440
 GET01450
 GET01460
 GET01470
 GET01480
 GET01490
 GET01500
 GET01510
 GET01520
 GET01530
 GET01540
 GET01550
 GET01560
 GET01570
 GET01580

FILE: GETST

```

      LOOPCT = NOCHAN/12
      LOOPCT = MOD(NOCHAN,12)
      IF (LOOPCT.GT. 0) LOOPCT = LOOPCT + 1
      IF (LOOPCT.EQ. 1) IEND = NOCHAN
      DO 240 I=1,LOOPCT
      START = ISTART
      IEND = IEND
      WRITE(6,210) (BLANK,CHNVEC(I),I=ISTART,IEND)
      FORMAT(2X,'CLUSTER',5X,12(A1,'CH(',12,')',2X))
      DO 230 J=1,NOSUR2
      WRITE(6,220) SURVEC(J), (MENS(K),K=START,END)
      FORMAT(5X,12,7X,12(F7.2,2X))
      START = ISTART + NOCHAN*J
      IEND = IEND + NOCHAN*J
230  CONTINUE
      WRITE(6,235)
      FORMAT(/)
      ISTART = IEND + 1
      IEND = IEND + ISTART - 1
      IF (IEND.GT. NOCHAN) IEND = NOCHAN
240  CONTINUE
      RETURN
      END

```

```
GET 01590
GET 01600
GET 01610
GET 01620
GET 01630
GET 01640
GET 01650
GET 01660
GET 01670
GET 01680
GET 01690
GET 01700
GET 01710
GET 01720
GET 01730
GET 01740
GET 01750
GET 01760
GET 01770
GET 01780
GET 01790
GET 01800
GET 01810
```

FILE: GRPSCN

```

C      FUNCTION GRPSCN(CARD,NNCLAS,GRPTR)
C      IMPLICIT INTEGER (A-H,O-Z)
C      -----
C      CALL..      CALL GRPSCN(CARD,NNCLAS,GRPTR)
C      ARGS..      CAPD   = 62 COL CARD BUFFER
C                  NNCLAS = MAX NO OF CLASSES TO ALLOW
C                  GRPTR  = PTR TO 'GROUPS'
C      REQUIRES..  COMMONS /INFORM/ /INFORS/ /DISCOM/
C                  ROUTINES NXTCHR  FIXUP  NUMBER
C      PURPOSE..   SCANS ALL 'GROUP' (TRAIN/TEST) CARDS
C                  AND SET UP 'GRPDEX','GRPNAM','GROUPS'
C      RETURNS..  GRPTR  - SEE ARGS
C      -----
C      INCLUDE COMR1.LIST
C      COMMON/INFORM/NOCLS2,NOSUB2,NOFET2,VARSZ2,TOTVT2,NOFLD2,
C      *      AVAR2,COVAR2,CLS1D2,SUBNO2,SUBDS2,FLOSV2,VERTX2,
C      *      FETVC2(30),SUBVC2(75),SUBPTR(75),CLSV2(60),
C      *      KEPPTS(60),NOGRP,GRPNAM(60),GRPDEX(61),
C      *      GRPCHK(61),GROUPS(124)
C      SEND
C      -----
C      DIMENSION BUF(4), CARD(62), COMVEC(2), NUMVEC(30)
C      LOGICAL*1 LCHAR(4)
C      DIMENSION ICHAR(1)
C      EQUIVALENCE (LCHAR(1),ICHAR(1))
C      LOGICAL*1 LLCHAR(4)
C      DIMENSION IICHAR(1)
C      EQUIVALENCE (LLCHAR(1),IICHAR(1))
C      DATA BLANK/' ', COMMA/',', COMVEC/1,0,0/
C      -----
C      COL = 0
C      GRPSCN = 1
C      J = NXTCHR(CARD,COL)
C      IF (J.EQ.BLANK) GO TO 110
C      DO 10 I=1,4
C      J2 = CARD(COL-1+I)
C      IF (J2.EQ.COMMA) GO TO 20
10  BUF(I) = J2
C      GO TO 40
C      20 DO 30 J=1,4
C      30 BUF(J) = BLANK
C      40 N = ABS(0.0)
C      DO 50 I=1,4
C      IICHAR(I)=BUF(I)
C      LCHAR(I)=LLCHAR(I)
50  WRD1=ICHAR(1)
C      GRPNAM(NOGRP+1) = WRD1
C      J = FIND12(CARD,COL,COMVEC)
C      IF (J.LE.0) GO TO 110
C      J = NUMBER(CARD,COL,NUMVEC,0)
C      II = 0
C      LAST = 0
C      DO 90 I=1,J
C      JJ = NUMVEC(I)
C      IF (JJ.GT.LAST.AND.JJ.LE.NNCLAS.AND.GRPCHK(JJ).EQ.0) GO TO 80
C      WRITE(6,70) JJ, JJ, CARD
70  FORMAT(// 5X, '///// FROM SUBR. GRPSCN --- CLASS ',15,
C      GRP00010
C      GRP00020
C      GRP00030
C      GRP00040
C      GRP00050
C      GRP00060
C      GRP00070
C      GRP00080
C      GRP00090
C      GRP00100
C      GRP00110
C      GRP00120
C      GRP00130
C      GRP00140
C      GRP00150
C      GRP00160
C      GRP00170
C      GRP00180
C      GRP00190
C      GRP00200
C      GRP00210
C      GRP00220
C      GRP00230
C      GRP00240
C      GRP00250
C      GRP00260
C      GRP00270
C      GRP00280
C      GRP00290
C      GRP00300
C      GRP00310
C      GRP00320
C      GRP00330
C      GRP00340
C      GRP00350
C      GRP00360
C      GRP00370
C      GRP00380
C      GRP00390
C      GRP00400
C      GRP00410
C      GRP00420
C      GRP00430
C      GRP00440
C      GRP00450
C      GRP00460
C      GRP00470
C      GRP00480
C      GRP00490
C      GRP00500
C      GRP00510
C      GRP00520
C      GRP00530
C      GRP00540
C      GRP00550
C      GRP00560
C      GRP00570
C      GRP00580
C      GRP00590
C      GRP00600
C      GRP00610
C      GRP00620
C      GRP00630
C      GRP00640
C      GRP00650
C      GRP00660
C      GRP00670
C      GRP00680
C      GRP00690
C      GRP00700
C      GRP00710
C      GRP00720
C      GRP00730
C      GRP00740
C      GRP00750
C      GRP00760
C      GRP00770
C      GRP00780
C      GRP00790

```

FILE: GRPSCN

```

1  INCORRECT --- CLASS '15' IGNORED // 11X CARD BEING SCANNED
2  IS .../9X,2H,, 62A1, 2H: /)
GO TO 90
90  II = 11
    NUMVEC(II) = JJ
    GRPCHK(JJ) = 1
    LAST = JJ
90  CONTINUE
    IF (II.LE.0) GO TO 110
C
    NOGRP = NOGRP+1
    GRPTR = GRPTR+1
    GRPDEX(NOGRP) = GRPTR
    GROUPS(GRPTR) = II
    DO 100 I=1,II
100  GROUPS(GRPTR+I) = NUMVEC(I)
    GRPTR = GRPTR+II
    GRPSCN = 0
    RETURN
C
110 RETURN
    END

```

GRP00800
 GRP00810
 GRP00820
 GRP00830
 GRP00840
 GRP00850
 GRP00860
 GRP00870
 GRP00880
 GRP00890
 GRP00900
 GRP00910
 GRP00920
 GRP00930
 GRP00940
 GRP00950
 GRP00960
 GRP00970
 GRP00980
 GRP00990
 GRP01000
 GRP01010
 GRP01020

FILE: HISTGM

```

SUBROUTINE HISTGM(FILHIS,FLOTAL,TOTAL)
.....
PURPOSE  -- CALCULATES HISTOGRAMS AND WRITES TOTAL HISTOGRAMMED
           STATISTICS ON UNIT 13 TO BE READ BY GRAYMP PROCESSOR
.....
IMPLICIT INTEGER(A-T)
INCLUDE COMMK3.LIST
INCLUDE COMMK4.LIST
INCLUDE COMMK6.LIST
COMMON /GHCRLK/MAXFET,NOFEAT,NOFET2,FETVEC(30),
      FETVC2(30),FLOINF(6),INFMT,FILESV,NOMIST,
      HISVEC(30),NOFLD,      FLOPTS
      *XSIZ,XLOW,XHGH,YSIZ
      *DIMENSION MED1(15),MED2(15),DATE(3),COMENT(15)
      *EQUIVALENCE (MED1(1),HEAD(4)),(DATE(1),HEAD(22)),
      *              (MED2(1),HEAD(30)),(COMENT(1),HEAD(48))
      *COMMON/GLOBAL/HEAD(63),MAPTAP,DATEPE,SAVTAP,BMFILE,HMKEY,
      *              HISFIL,HISKEY,TRFORM,ERITP,ERPKEY,MAPUNT,NOFILE,
      *              DRUMAD,DHMDUS,PAGSIZ,DATEFIL,STAFIL,ASAV,ASAVFL
      *              ,NHSTUN,NHSTFI,SCTRUN,MAPFIL
      *              ,DOTUNT,DOTFIL,NCMPAS,TRNSFL,BMTRFL,HISTFL,PCHUNT,
      *              CRDUNT,PRUNT,RANDIO
CSEND
COMMON /HISTOR/HF
C
INTEGER XSIZ,YSIZ,XHGH,XLOW
INTEGER VERTCS
DIMENSION DATA(12000),FILHIS(NOFEAT,256),
      *FLOTAL(NOMIST,XSIZ),TOTAL(NOMIST,XSIZ),IFLD(50,24),
      *VERTCS(2,11),FL(4)
      *DATA BLANK/1/2/DIM/12000/
      *DATA TOTAL/1/TOTAL/
      *EQUIVALENCE (FLDINF(1),LINSTR),(FLDINF(2),LINEND),
      *              (FLDINF(3),LININC),(FLDINF(4),SAMSTR),
      *              (FLDINF(5),SAMEND),(FLDINF(6),SAMINC)
      *CALL TAPHOR(DATEPE,DATEFIL)
      *HISKEY=1
      *FILESV = 0
      *ISWTH = 0
      *DO 7 J=1,XSIZ
      *DO 7 I=1,NOMIST
      *FLOTAL(I,J) = 0
      *TOTAL(I,J) = 0
7 TOTAL(I,J) = 0
NOFLD=0
IF(HF.NE.1) GO TO 10
NC=5
VERTCS(1,1)=1
VERTCS(2,1)=1
VERTCS(1,2)=200
VERTCS(2,2)=1
VERTCS(1,3)=200
VERTCS(2,3)=500
VERTCS(1,4)=1
VERTCS(2,4)=500
VERTCS(1,5)=1
VERTCS(2,5)=1
FLDNAM=BLANK
FLDINF(1)=1
FLDINF(2)=500
FLDINF(3)=10
FLDINF(4)=1
FLDINF(5)=200
FLDINF(6)=10
GO TO 15
C
READ IN FIELD CARDS
10 ICK=1,AREAD(FLDNAM,VERTCS,FLDINF,NC)
IF(ICK.EQ.1) GO TO 15
IF(ICK.EQ.0) GO TO 60
IF(ICK.LE.-1) GO TO 10
15 NSAMP = ( FLDINF(5) - FLDINF(4) ) / FLDINF(6) + 1
CHECK DATA DIMENSIONS. IF TOO MUCH DATA REQUESTED, RESET SAMPLE END
TOTPTS = NSAMP*NOFEAT

```

FILE: HISTGM

```

      IF (TOTPTS .LE. DIM) GO TO 14
      NSAMP = DIM / NOFEAT
      FLDINF(5) = (NSAMP-1)*FLDINF(6) + FLDINF(4)
      WRITE(6,300) NSAMP
300  FORMAT(' TOO MUCH DATA REQUESTED -- SAMPLE END WAS RESET TO:15/')
C
C  STORE FIELD INFORMATION
C
14  NOFLD = NOFLD + 1
      IF (NOFLD.GT.50) FILES=-2
      IF (NOFLD.GT.50) GO TO 19
      K=0
      DO 100 J=1,10
      DO 100 I=1,2
      KK=K+1
100  FLD(NOFLD,K)=VERTCS(I,J)
      FLD(NOFLD,21)=FLDNAM
      FLD(NOFLD,22)=SAMINC
      FLD(NOFLD,23)=LININC
      FLD(NOFLD,24)=NC
19  CONTINUE
      ISWTH = ISWTH + 1
C
C  ZERO OUT PART OF FIELD HISTOGRAM ARRAY
C
      DO 20 I=1,NOFEAT
      DO 20 J=1,256
20  FILHIS(I,J) = 0
C
C  SCALING FACTORS USED FOR PLOTTING ROUTINE
C
      XSCALE = FLOAT(1-XSIZ)/(XMGH-XLOW)
      XSMFT = -XMGH*XSCALE + 1.0
C
C  PRINT FIELD STATS
C
      CALL FLDINT(FLDINF,FETVEC,NOFEAT)
      LINES = (FLDINF(2)-FLDINF(1)) / FLDINF(3) + 1
      FLDPPTS=0
      DO 30 I=1,LINES
      N = 1
      CALL LINERD(I,DATA,ENDTAP)
      IF (ENDTAP .NE. 0) GO TO 33
      IF (I.NE.1) GO TO 101
      II=LINSTH
      GO TO 102
101  II=II+LININC
102  CALL FDLINT(VERTCS,NC ,FL,II,NS,JJ)
      KC=0
      DO 30 J=1,NOFEAT
      L=1
      DO 34 KK=1,NSAMP
      KPT=(KK-1)*SAMINC+SAMSTR
      DO 103 JK=L,JJ,2
      IF (KPT.LT.FL(JK)) GO TO 34
      IF (KPT.GT.FL(JK+1)) GO TO 105
      IF (J.EQ.1) FLDPPTS=FLDPPTS+1
      IPOS = (J-1)*NSAMP + KK
      K = IDATA(IPOS)
      IF (K .LT. 1) K = 1
      FILHIS(J,K) = FILHIS(J,K) + 1
      IF (HISVEC(N).NE.FETVEC(J)) GO TO 34
      KC=1
      IPUT = IDATA(IPOS) * XSCALE + XSMFT + 0.501
      IF (IPUT .LT. 1) IPUT = 1
      IF (IPUT .GT. XSIZ) IPUT = XSIZ
      FLDTAL(N,IPUT) = FLDTAL(N,IPUT) + 1
      TOTTAL(N,IPUT) = 1 + TOTTAL(N,IPUT)
      IF (KK.EQ. NSAMP) N = N + 1
      GO TO 34
103  CONTINUE
105  L=L+2
      IF (L.GT. JJ) GO TO 106
106  CONTINUE
107  CONTINUE
      IF (KK.EQ. NSAMP) GO TO 30
104  CONTINUE
      IF (KC.EQ.1) N=N+1
      KC=0
30  CONTINUE

```

```

111 000800
111 000810
111 000820
111 000830
111 000840
111 000850
111 000860
111 000870
111 000880
111 000890
111 000900
111 000910
111 000920
111 000930
111 000940
111 000950
111 000960
111 000970
111 000980
111 000990
111 001000
111 001010
111 001020
111 001030
111 001040
111 001050
111 001060
111 001070
111 001080
111 001090
111 001100
111 001110
111 001120
111 001130
111 001140
111 001150
111 001160
111 001170
111 001180
111 001190
111 001200
111 001210
111 001220
111 001230
111 001240
111 001250
111 001260
111 001270
111 001280
111 001290
111 001300
111 001310
111 001320
111 001330
111 001340
111 001350
111 001360
111 001370
111 001380
111 001390
111 001400
111 001410
111 001420
111 001430
111 001440
111 001450
111 001460
111 001470
111 001480
111 001490
111 001500
111 001510
111 001520
111 001530
111 001540
111 001550
111 001560
111 001570
111 001580

```

FILE: HISTOM

```

33  CONTINUE
    IP = 0
    CALL HISTIC(FILHIS,IP,IFLD,      VERTCS,NC)
    WRITE(HISFIL) ((FILHIS(I,J),J=1,256),I=1,NOFEAT)
    IF (NF.FO.1) GO TO 60
    IF (NOMIST.EQ.0) GO TO 10
C
    CALL PLOTTING ROUTINE TO PLOT HISTOGRAM FOR THE FIELDS
C
    CALL HISTGRM(FLOTAL,IDATA,FLDNAM,2,XSIZ,XHGH,XLOW,YSIZ,
    *NOMIST,FLOPTS,HISVEC)
    GO TO 10
C
    WRITE TOTAL HIST ON TAPE --UNIT 13
C
60  CONTINUE
    REWIND HISFIL
    DO 104 I=1,NOFEAT
    DO 104 J=1,256
104  FILHIS(I,J)=0
    DO 107 K=1,NOFLD
    JK=NOFEAT*256
    READ(HISFIL) (IDATA(I),I=1,JK)
    M=0
    DO 108 I=1,NOFEAT
    DO 108 J=1,256
    M=M+1
108  FILHIS(I,J)=IDATA(M)+FILHIS(I,J)
107  CONTINUE
    REWIND HISFIL
    WRITE(HISFIL) NOFEAT, (FETVEC(I),I=1,NOFEAT)
    WRITE(HISFIL) ((FILHIS(I,J),J=1,256),I=1,NOFEAT)
    REWIND HISFIL
C
    PRINT TOTAL STATS
C
    IP = -1
    CALL HISTIC(FILHIS,IP,IFLD,      VERTCS,NC)
    IF (NOMIST.EQ.0) RETURN
    IF (NOFLD.EQ.1) RETURN
C
    CALL PLOTTING ROUTINE TO PLOT TOTAL HISTOGRAM
C
    CALL HISTGRM(TOTAL,IDATA,TOTAL,3,XSIZ,XHGH,XLOW,YSIZ,
    *NOMIST,FLOPTS,HISVEC)
    RETURN
C 200  FORMAT(13A6,A2)
    END

```

```

H1501590
H1501600
H1501610
H1501620
H1501630
H1501640
H1501650
H1501660
H1501670
H1501680
H1501690
H1501700
H1501710
H1501720
H1501730
H1501740
H1501750
H1501760
H1501770
H1501780
H1501790
H1501800
H1501810
H1501820
H1501830
H1501840
H1501850
H1501860
H1501870
H1501880
H1501890
H1501900
H1501910
H1501920
H1501930
H1501940
H1501950
H1501960
H1501970
H1501980
H1501990
H1502000
H1502010
H1502020
H1502030
H1502040
H1502050
H1502060

```

FILE: HISTIC

```

C ..... SURROUTINE HISTIC(IMG,NI,IFLD, VERTCS,NC) ..... X 500010
C ..... PURPOSE -- COMPUTES AND DISPLAYS STATS FOR HISTOGRAM ROUTINE ..... X 500020
C ..... INCLUDE COMH3.LIST ..... X 500030
C ..... INCLUDE COMH4.LIST ..... X 500040
C ..... INCLUDE COMH4.LIST ..... X 500050
C ..... COMMON /GRCHLR,MAXFET,NOFEAT,NOFET2,FETVEC(30), ..... X 500060
C ..... FETVC2(30),FLDINF(6),INFMT,FILESV,NOHIST, ..... X 500070
C ..... HISVEC(30),NOFLD, FLOPTS ..... X 500080
C ..... XSIZE,XLOW,XHIGH,YSIZ ..... X 500090
C ..... DIMENSION HED1(15),HED2(15),DATE(3),COMENT(15) ..... X 500100
C ..... EQUIVALENCE (HED1(1),HEAD(4)),(DATE(1),HEAD(22)), ..... X 500110
C ..... (HED2(1),HEAD(30)),(COMENT(1),HEAD(48)) ..... X 500120
C ..... COMMON/GLOBAL/HEAD(63),MAPTAP,DATAPE,SAVTAP,BMFILE,BMKEY, ..... X 500130
C ..... HISFIL,HISKEY,IRFORM,ERIPTP,ERPKEY,MAPUNT,NOFILE, ..... X 500140
C ..... DRUMAD,DRMUDS,PAGSIZ,DATFIL,STAFIL,ASAV,ASAVFL ..... X 500150
C ..... ,NHSTUN,NHSTFI,SCTHUN,MAPFIL ..... X 500160
C ..... ,DOTUNT,DOTFIL,NCHPAS,TRNSFL,BMTRFL,HISTFL,PCHUNT, ..... X 500170
C ..... CRDUNT,PRUNT,RANDIO ..... X 500180
CSEND ..... X 500190
C ..... DIMENSION RANGE(30,2),ZMEAN(30), STDDEV(30), ..... X 500200
C ..... ,NRANGE(30,2),IMG(NOFEAT,256),IFLD(50,24) ..... X 500210
C ..... INTEGER VERTCS(1),OP,CP,COMMA ..... X 500220
C ..... DATA OP/1,2,CP/1,2,COMMA/1,2 ..... X 500230
C ..... INTEGER FLDINF,FETVEC,HEAD,FILESV ..... X 500240
C ..... REAL NRANGE ..... X 500250
C ..... COMPUTES THE DATA RANGE ..... X 500260
C ..... DO 10 J = 1,NOFEAT ..... X 500270
C ..... DO 20 J = 1,256 ..... X 500280
C ..... ISTR = J ..... X 500290
C ..... IF (IMG(I,J) .NE. 0) GO TO 30 ..... X 500300
C ..... CONTINUE ..... X 500310
C ..... 20 RANGE(I,1) = J ..... X 500320
C ..... K=257 ..... X 500330
C ..... 40 K=K-1 ..... X 500340
C ..... IF (IMG(I,K) .NE. 0) GO TO 50. ..... X 500350
C ..... IF (K.GT.ISTR) GO TO 40 ..... X 500360
C ..... 50 RANGE(I,2) = K ..... X 500370
C ..... COMPUTES THE MEAN -- (1/N) * ( 1*IMG(1) + 2*IMG(2) + ... N*IMG(N) ) ..... X 500380
C ..... STDDEV1 = 0 ..... X 500390
C ..... RMEAN = 0 ..... X 500400
C ..... N = 0 ..... X 500410
C ..... DO 60 L = 1,256 ..... X 500420
C ..... N = IMG(I,L) * N ..... X 500430
C ..... AMEAN = L*IMG(I,L) ..... X 500440
C ..... STDDEV = AMEAN * L ..... X 500450
C ..... RMEAN = AMEAN * RMEAN ..... X 500460
C ..... 60 STDDEV1 = STDDEV * STDDEV1 ..... X 500470
C ..... ZMEAN(I) = RMEAN / N ..... X 500480
C ..... STDDEV1 = STDDEV1 / N ..... X 500490
C ..... COMPUTES THE STD. DEV. -- SORT.( (1/N)*(1**2 * IMG(1) + 2**2 * IMG( ..... X 500500
C ..... 2) * ... * N**2 * IMG(N)) - MEAN** ..... X 500510
C ..... 2 ) ..... X 500520
C ..... A ..... X 500530
C ..... = STDDEV1 - ZMEAN(I)**2 ..... X 500540
C ..... STDDEV(I) = SORT(A) ..... X 500550
C ..... COMPUTES NORMALIZED RANGE ..... X 500560
C ..... NRANGE(I,1) = ZMEAN(I) - 3*STDDEV(I) ..... X 500570
C ..... 10 NRANGE(I,2) = ZMEAN(I) + 3*STDDEV(I) ..... X 500580
C ..... IN=NC-1 ..... X 500590
C ..... NNC=2*(IN) ..... X 500600
C ..... WRITE(6,HEAD) ..... X 500610
C ..... IF (N.FO. -1) GO TO A5 ..... X 500620
C ..... WRITE(6,510) ..... X 500630
C ..... DO 70 I1 = 1,NOFEAT ..... X 500640
C ..... 70 WRITE(6,520) FETVEC(I1),IFLD(NOFLD,21),IN,FLDINF(6),FLDINF(3), ..... X 500650
C ..... , (10P,V,VERTCS(I1),COMMA,VERTCS(I1+1),CP),J=1,NNC,2) ..... X 500660
C ..... WRITE(6,530) ..... X 500670
C ..... 75 CONTINUE ..... X 500680
C ..... X 500690
C ..... X 500700
C ..... X 500710
C ..... X 500720
C ..... X 500730
C ..... X 500740
C ..... X 500750
C ..... X 500760
C ..... X 500770
C ..... X 500780
C ..... X 500790

```

FILE: HISTIC

```

DO 80      I3 = 1,NOFEAT
80 WRITE(4,540) FETVEC(I3),RANGE(I3,1),RANGE(I3,2),ZMEAN(I3),
*STDEV(I3),NRANGE(I3,1),NRANGE(I3,2)
RETURN
85 WRITE(6,550)
IF(NOFLO.GT.50) NOFLD=50
DO 1 I=1,NOFLD
IN=IFLD(I,24)-1
NMC=IN*2
KJ=10
IF(NMC.LE.10) KJ=NMC
DO 2 J=1,NMC
2 VERTCS(J)=IFLD(I,J)
1 WRITE(6,560) IFLD(I,21),IN,IFLD(I,22),IFLD(I,23),
*((OP,VERTCS(K),COMMA,VERTCS(K+1),CP),K=1,KJ,2)
IF(NMC.LE.10) GO TO 2017
WRITE(6,561) ((OP,VERTCS(K),COMMA,VERTCS(K+1),CP),K=11,NMC,2)
2017 CONTINUE
IF (FILESV.EQ. -2) WRITE(6,565)
WRITE(6,570)
GO TO 75
530 FORMAT(//////// T53,
* 'HISTOGRAM STATISTICS'// 16X,'CHANNEL',9X,'DATA',
* 'RANGE',9X,'MEAN', 7X,'STANDARD DEVIATION', 9X,'NORMALIZED',
* 'RANGE'// 86X,'(MEAN + AND - 3 STD DEV)'//)
540 FORMAT(18X,I2,11X,F5.1,2X,F5.1,7X,F5.1,12X,F5.1,17X,F6.1,2X,F6.1)
510 FORMAT(////////T53,'DATA BLOCK(S) HISTOGRAMMED'//,
* T24,'NO. OF SAMPLE LINE'//
* T3,'CHANNEL FIELDNAME VERTICES INC INC VERTICES(SAMPLE,LI
* NE))
550 FORMAT(////////T53,'DATA BLOCK(S) HISTOGRAMMED'//,
* T24,'NO. OF SAMPLE LINE'//
* T12,'FIELDNAME VERTICES INC INC VERTICES(SAMPLE,LI
* NE))
520 FORMAT(14X,I2,7X,A4,8X,I2,7X,I4,2X,I4,1X,
* 5(A1,I4,A1,I4,A1,2X)/T46,5(A1,I4,A1,I4,A1,2X))
540 FORMAT(13X,A4,8X,I2,7X,I4,2X,I4,1X,
* 5(A1,I4,A1,I4,A1,2X))
541 FORMAT(T46,5(A1,I4,A1,I4,A1,2X))
565 FORMAT(T2,'ONLY THE FIRST 50 FIELD DESCRIPTIONS WERE PRINTED,
* BUT ALL THE FIELDS WERE INCLUDED IN THE TOTAL HISTOGRAMMED STATS'
*)
570 FORMAT(//////// T60,'TOTAL'// T53,
* 'HISTOGRAM STATISTICS'// 16X,'CHANNEL',9X,'DATA',
* 'RANGE',9X,'MEAN', 7X,'STANDARD DEVIATION', 9X,'NORMALIZED',
* 'RANGE'// 86X,'(MEAN + AND - 3 STD DEV)'//)
END

```

PRECEDING PAGE NOT BLANK - MISHNUMBER.

FILE: I4A1BN

```

C      SURROUTINE I4A1BN(IFLD,NCHFLD,NCVTED)
C      DAVID LFF SMITH 9 SEPTEMBER 1977.
C      THIS SURROUTINE ACCEPTS AN ARRAY OF EBCDIC CHARACTERS AND CONVERTS
C      EBCDIC DIGITS TO A BINARY INTEGER.
C
C      CALLING SEQUENCE:
C      CALL I4A1BN( FIELD, LENGTH, OUTPUT )
C      "WHERE FIELD IS THE FIRST WORD OF AN ARRAY OF EBCDIC CHARACTERS
C      TO BE CONVERTED TO BINARY. CHARACTERS STORED ONE PER
C      WORD, LEFT JUSTIFIED, AS BY AN A1 FORMAT.
C      LENGTH IS THE NUMBER OF CHARACTERS IN THE FIELD.
C      AND OUTPUT IS THE ONE WORD RESULT.
C
      INTEGER * 4 IDUM(2), IFLD(20)
      LOGICAL * 1 L(8)
      EQUIVALENCE (L(1),IDUM(1)),(ILCH,IDUM(1)),(ICHAR,IDUM(2))
      DATA ICHAR / 0 /
      DATA IR0 / 240 /
      DATA IR9 / 249 /
      DATA IRBL / 64 /
      DATA IRPL / 78 /
      DATA IRMI / 96 /
      NCVTED = 0
      IERFLG = 0
      MINUS = 1
      DO 30 I = 1, NCHFLD
        ILCH = IFLD( I )
        L(1) = L(1)
        IF ( ICHAR .LT. IR0 ) GO TO 10
        IF ( ICHAR .GT. IR9 ) GO TO 10
        JDIG = I
        GO TO 200
      10  NEXT = I + 1
        IF ( ICHAR .EQ. IRBL ) GO TO 30
        IF ( ICHAR .EQ. IRPL ) GO TO 100
        IF ( ICHAR .NE. IRMI ) GO TO 20
        MINUS = - MINUS
        GO TO 100
      20  IERFLG = I
        CONTINUE
      30  IERFLG = NCHFLD + 1
        GO TO 240
      100 IF ( NEXT .GT. NCHFLD ) GO TO 130
        DO 120 I = NEXT, NCHFLD
          ILCH = IFLD( I )
          L(1) = L(1)
          IF ( ICHAR .LT. IR0 ) GO TO 110
          IF ( ICHAR .GT. IR9 ) GO TO 110
          JDIG = I
          GO TO 200
        110 IF ( ICHAR .EQ. IRBL ) GO TO 120
          IERFLG = I
          CONTINUE
        120 CONTINUE
        130 IERFLG = NCHFLD + 1
        GO TO 240
      200 DO 230 I = JDIG, NCHFLD
          ILCH = IFLD( I )
          L(1) = L(1)
          IF ( ICHAR .LT. IR0 ) GO TO 210
          IF ( ICHAR .LE. IR9 ) GO TO 220
          IF ( ICHAR .NE. IRBL ) IERFLG = I
          ICHAR = IR0
        210 IVAL = ICHAR - IR0
          NCVTED = NCVTED * 10 - IVAL
        220 CONTINUE
        230 IF ( MINUS .EQ. 1 ) NCVTED = - NCVTED
        240 IF ( IERFLG .EQ. 0 ) GO TO 250
          NCH = NCHFLD
          IF ( NCH .GT. 40 ) NCH = 40
          WRITE ( 6, 1000 ) IERFLG, NCHFLD, (IFLD(K), K = 1, NCH )
          FORMAT ( ' EBCDIC TO BINARY INTEGER CONVERSION ERROR'// ' AT CHARACTER'
1000 1 ' I, IS, ' OF ' I5, ' CHARACTER FIELD: '/(X,80A1)
        250 RETURN
      END

```

I4A00010
 I4A00020
 I4A00030
 I4A00040
 I4A00050
 I4A00060
 I4A00070
 I4A00080
 I4A00090
 I4A00100
 I4A00110
 I4A00120
 I4A00130
 I4A00140
 I4A00150
 I4A00160
 I4A00170
 I4A00180
 I4A00190
 I4A00200
 I4A00210
 I4A00220
 I4A00230
 I4A00240
 I4A00250
 I4A00260
 I4A00270
 I4A00280
 I4A00290
 I4A00300
 I4A00310
 I4A00320
 I4A00330
 I4A00340
 I4A00350
 I4A00360
 I4A00370
 I4A00380
 I4A00390
 I4A00400
 I4A00410
 I4A00420
 I4A00430
 I4A00440
 I4A00450
 I4A00460
 I4A00470
 I4A00480
 I4A00490
 I4A00500
 I4A00510
 I4A00520
 I4A00530
 I4A00540
 I4A00550
 I4A00560
 I4A00570
 I4A00580
 I4A00590
 I4A00600
 I4A00610
 I4A00620
 I4A00630
 I4A00640
 I4A00650
 I4A00660
 I4A00670
 I4A00680
 I4A00690
 I4A00700
 I4A00710
 I4A00720
 I4A00730

FILE: LARMAN

```
SUBROUTINE LARMAN(UNIT,FILE,NOCLS,TOTSUB,NOFEAT,TOTFLD,TOTVRT,  
FETVEC,FLDSAV,VERTEX,CLSNMS,NOSUBS,SUBNM,N,  
STADRS,VARSIZ,PUNCH,SUBVEC,PRNSTS,SWICH)  
.....  
SUBROUTINE STATFL WILL WRITE A STATISTICS FILE IN THE FORMAT  
EXPECTED BY THE LARSYS PROGRAM. IT WILL ALSO PUNCH THE MODULE  
DECK IF REQUESTED.  
ARGUMENTS ARE ALL INPUT.  
UNIT - FORTRAN UNIT NUMBER WHERE THE FILE IS TO BE WRITTEN.  
FILE - FILE NUMBER ON 'UNIT' WHERE THE FILE IS TO BE WRITTEN.  
NOCLS - NUMBER OF CLASSES.  
TOTSUR - TOTAL NUMBER OF SUBCLASSES FOR ALL CLASSES.  
NOFEAT - NUMBER OF CHANNELS.  
TOTFLD - TOTAL NUMBER OF TRAINING FIELDS.  
TOTVRT - TOTAL NUMBER OF VERTICES FOR ALL TRAINING FIELDS.  
FETVEC - VECTOR CONTAINING THE CHANNEL NUMBERS FOR WHICH  
THE STATISTICS WERE COMPUTED. (DIMENSIONED - NOFEAT)  
FLDSAV - ARRAY DIMENSIONED - 4 BY TOTFLD AND CONTAINING THE  
FOLLOWING INFORMATION FOR EACH TRAINING FIELD.  
FLDSAV(1,I) - NAME OF FIELD I.  
FLDSAV(2,I) - CLASS NO. THAT FIELD I BELONGS TO.  
FLDSAV(3,I) - SUBCLASS NO. THAT FIELD I BELONGS TO.  
ZERO IF THE FIELD IS NOT ASSOCIATED  
WITH A SUBCLASS.  
FLDSAV(4,I) - NO. OF VERTICES FOR THIS FIELD,  
INCLUDING THE CLOSURE POINT.  
VERTEX - ARRAY CONTAINING VERTICES FROM ALL TRAINING FIELDS.  
DIMENSIONED - 2*TOTVRT, THE VERTICES SHOULD BE STORED  
SAMPLE NO. FIRST THEN LINE NUMBER FOR EACH VERTEX.  
CLOSURE POINTS MUST BE INCLUDED FOR EACH VERTEX.  
(I.E. THE FIRST VERTEX IS REPEATED AS THE LAST VERTEX)  
CLSNMS - ARRAY CONTAINING ALPHANUMERIC CLASS NAMES.  
NOSUBS - ARRAY CONTAINING THE NO. OF SUBCLASSES IN EACH CLASS.  
SUBNM - ARRAY CONTAINING ALPHANUMERIC SUBCLASS NAMES.  
N - ARRAY CONTAINING THE NO. OF PIXELS IN EACH SUBCLASS.  
PUNCH - IF PUNCH=1 THE MODULE DECK WILL BE PUNCHED.  
SWICH = 1 CALLED BY ISOCLS  
SWICH = 2 CALLED BY LABEL  
SWICH = 3 CALLED BY STAT  
SWICH = 4 CALLED BY TRSTAT  
.....  
IMPLICIT INTEGER(A-Z)  
DIMENSION FETVEC(NOFEAT),FLDSAV(4,TOTFLD),VERTEX(TOTVRT)  
DIMENSION CLSNMS(NOCLS),NOSUBS(NOCLS),SUBNM(TOTSUB),N(TOTSUR)  
DIMENSION SURVEC(1)  
DATA PCHUNT /7/  
REAL COVAR(465),MEANS(30)  
POSITION 'UNIT' TO CORRECT FILE NO.  
SAVTAP=UNIT  
REWIND SAVTAP  
NF = FILE  
CALL FSHSFL(SAVTAP,NF,ISTAT1)  
IF(ISTAT1.EQ.0)GO TO 1  
WRITE(6,220)FILE  
220 FORMAT(/15,'ERROR IN POSITIONING SIG. EXTENSION TAPE TO FILE',13/  
'15,'OUTPUT FILE NOT WRITTEN')  
1 CONTINUE  
IR=1  
DATA PCHUNT/7/  
IF(PUNCH.NE.1)GO TO 6  
WRITE(PCHUNT,110)  
WRITE(PCHUNT,120)NOCLS,TOTSUB,NOFEAT,TOTFLD,TOTVRT  
WRITE(PCHUNT,130)(FETVEC(I),I=1,NOFEAT)  
6 CONTINUE  
IF(ISTAT1.NE.0) GO TO 11  
WRITE(SAVTAP)NOCLS,TOTSUB,NOFEAT,TOTFLD,TOTVRT,  
(FETVEC(J),J=1,NOFEAT)  
11 CONTINUE  
DO 2 I=1,TOTFLD  
NV = FLDSAV(4,I)
```

FILE: LARMAN

```

IF=IR *NV*2 -1
IF (PUNCH.NE.1) GO TO 7
WRITE (PCHUNT,135) (FLDSAV(J,I),J=1,4)
WRITE (PCHUNT,140) (VERTEX(J),J=18,1E)
7 CONTINUE
IF (ISTAT1.NE.0) GO TO 3
WRITE (SAVTAP) (FLDSAV(J,I),J=1,4)
WRITE (SAVTAP) (VERTEX(J),J=18,1E)
3 IR=IF+1
2 CONTINUE
IF (PUNCH.NE.1) GO TO 8
WRITE (PCHUNT,145) (CLSNMS(I),I=1,NOCLS)
WRITE (PCHUNT,150) (NOSUBS(I),I=1,NOCLS)
WRITE (PCHUNT,155) (SUBNM(I),I=1,TOTSUB)
8 CONTINUE
IF (ISTAT1.NE.0) GO TO 18
WRITE (SAVTAP) (CLSNMS(I),I=1,NOCLS), (NOSUBS(I),I=1,NOCLS),
* (SUBNM(I),I=1,TOTSUB)
18 CONTINUE
III = 0
WRITE (6,225)
225 FORMAT(1H1)
NUMSUB = 0
TOTSTA = 0
DO 20 ICLAS=1,NOCLS
TOTSTA = NUMSUB + TOTSTA
NUMSUB=NOSUBS(ICLAS)
DO 20 J=1,NUMSUB
III=III+1
C STATS ARE COMING FROM ISOCLS
IF (SWTCH.NE.1) GO TO 50
MEAN1 = STADRS + (VARSI*NOFEAT)*TOTSTA
MEAN2 = MEAN1 + (J-1)*NOFEAT
COVAR1 = MEAN1 + NUMSUB*NOFEAT + VARSIZ*(J-1)
KK=III
GO TO 80
C STAT ARE COMING LABEL
50 IF (SWTCH.NE.2) GO TO 60
KK = SUBVEC(III)
MEAN1 = STADRS + VARSIZ*TOTSUB
MEAN2 = MEAN1 + NOFEAT*(KK-1)
COVAR1 = STADRS + VARSIZ*(KK-1)
GO TO 80
C STATS ARE COMING FROM STAT
60 CONTINUE
GO TO 80
C STATS ARE COMING TRSTAT
80 CONTINUE
C
C READ MEANS AND COVARIANCES INTO CORE FROM DRUM
C
CALL RREAD(COVAR1,COVAR2,VARSI,ISTAT)
30 IF (ISTAT.EQ.1) GO TO 30
CALL RREAD(MEAN2,MEANS,NOFEAT,ISTAT1)
40 IF (ISTAT1.EQ.1) GO TO 40
C
IF (PUNCH.NE.1) GO TO 9
WRITE (PCHUNT,170) N(KK)
WRITE (PCHUNT,90) (MEANS(K),K=1,NOFEAT)
WRITE (PCHUNT,100) (COVAR(K),K=1,VARSI)
9 CONTINUE
IF (ISTAT1.NE.0) GO TO 20
WRITE (SAVTAP) N(KK), (COVAR(K),K=1,VARSI), (MEANS(K),K=1,NOFEAT)
IF (PRNSTS.NE.1) GOTO 20
C
C PRINTS THE STATS ON THE LINE PRINTER
C
DATA RCDTWO/'2'/
WRITE (6,65)
65 FORMAT(//)
WRITE (6,310) CLSNMS(ICLAS), SUBNM(III)
310 FORMAT(// ' CLASS : 'A6// ' SUBCLASS: 'A6)
DO 340 LOC=1,NOFEAT,12
STOP=LOC+11
IF (STOP.GT.NOFEAT) STOP=NOFEAT
340 WRITE (6,350) (MEANS(I),I=LOC,STOP)
350 FORMAT(// ' MEAN: '3X,12F9.2)
WRITE (6,360)
360 FORMAT(// ' COVARIANCE MATRIX: ')

```

LAH00800
 LAH00810
 LAH00820
 LAH00830
 LAH00840
 LAH00850
 LAH00860
 LAH00870
 LAH00880
 LAH00890
 LAH00900
 LAH00910
 LAH00920
 LAH00930
 LAH00940
 LAH00950
 LAH00960
 LAH00970
 LAH00980
 LAH00990
 LAH01000
 LAH01010
 LAH01020
 LAH01030
 LAH01040
 LAH01050
 LAH01060
 LAH01070
 LAH01080
 LAH01090
 LAH01100
 LAH01110
 LAH01120
 LAH01130
 LAH01140
 LAH01150
 LAH01160
 LAH01170
 LAH01180
 LAH01190
 LAH01200
 LAH01210
 LAH01220
 LAH01230
 LAH01240
 LAH01250
 LAH01260
 LAH01270
 LAH01280
 LAH01290
 LAH01300
 LAH01310
 LAH01320
 LAH01330
 LAH01340
 LAH01350
 LAH01360
 LAH01370
 LAH01380
 LAH01390
 LAH01400
 LAH01410
 LAH01420
 LAH01430
 LAH01440
 LAH01450
 LAH01460
 LAH01470
 LAH01480
 LAH01490
 LAH01500
 LAH01510
 LAH01520
 LAH01530
 LAH01540
 LAH01550
 LAH01560
 LAH01570
 LAH01580

FILE: LABMAN

| | | |
|-----|--|----------|
| 20 | CALL WRTMTX(COVAR(1),NOFEAT,BCDTWO) | LAB01590 |
| | CONTINUE | LAB01600 |
| | IF (ISTAT1.NE. 0) GO TO 19 | LAB01610 |
| 19 | ENDFILE SAVTAP | LAB01620 |
| | CONTINUE | LAB01630 |
| | WRITE(6,180)NOCLS,TOTSUB | LAB01640 |
| | WRITE(6,190) | LAB01650 |
| | K=0 | LAB01660 |
| | J=1 | LAB01670 |
| 21 | NSURS=NOSUBS(I) | LAB01680 |
| | WRITE(6,205) | LAB01690 |
| | WRITE(6,200)I,CLSNMS(I),NSUBS | LAB01700 |
| | WRITE(6,205) | LAB01710 |
| | DO 25 J=1,NSUBS | LAB01720 |
| | K=K+1 | LAB01730 |
| | WRITE(6,210)K,SUBNM(K) | LAB01740 |
| 25 | CONTINUE | LAB01750 |
| | J=J+1 | LAB01760 |
| | IF(I.LF.NOCLS)GO TO 21 | LAB01770 |
| | RETURN | LAB01780 |
| 90 | FORMAT('MEAN ',5F15.8) | LAB01790 |
| 100 | FORMAT('COVAR',5F15.8) | LAB01800 |
| 110 | FORMAT('MODULE DECK FROM SLABEL') | LAB01810 |
| 120 | FORMAT('NOCLS ',I4,' NOSUB ',I4,' NOFEAT ',I3,' NOFLD ',I4,' TOTVRT ', | LAB01820 |
| | * I5) | LAB01830 |
| 130 | FORMAT('VECTR ',5X,30I2) | LAB01840 |
| 135 | FORMAT(A6,4X,3(I2,8X)) | LAB01850 |
| 140 | FORMAT('VERTICES ',14I5) | LAB01860 |
| 145 | FORMAT('CLSDESC ',9(2X,A6))) | LAB01870 |
| 150 | FORMAT('NOSUBS ',24(1X,I2))) | LAB01880 |
| 155 | FORMAT('SUBDESC ',10(A6,1X))) | LAB01890 |
| 160 | FORMAT('FREQ ',12F6.2) | LAB01900 |
| 170 | FORMAT('NOPTS ',7X,I8) | LAB01910 |
| 180 | FORMAT('/// THE STATISTICS FILE FOR ',I4,' CLASSES AND ',I4, | LAB01920 |
| | * ' SUBCLASSES HAS BEEN WRITTEN'//) | LAB01930 |
| 185 | FORMAT('H ', 'THE STATS WERE WRITTEN ON FILE ',I3) | LAB01940 |
| 190 | FORMAT(' THE STATS FOR A PARTICULAR CLASS OR SUBCLASS SHOULD BE RE | LAB01950 |
| | *FERRED TO IN LATER RUNS BY '/' THE FOLLOWING NAMES AND NUMBERS (W | LAB01960 |
| | *HICHEVER APPLICABLE)'/) | LAB01970 |
| 200 | FORMAT(5X,'CLASS ',I3,2X,A6,5X,' SUBCLASSES (TOTAL= ',I3,')') | LAB01980 |
| 205 | FORMAT('/') | LAB01990 |
| 210 | FORMAT(25X,I3,2X,A6) | LAB02000 |
| | END | LAB02010 |

ORIGINAL PAGE IS
OF POOR QUALITY

FILE: LAREAD

```

FUNCTION LAREAD(FLDNAM,VERTCS,FLDINF,NC)
  IMPLICIT INTEGER(A-Z)
  DIMENSION CARD(62), FLDINF(6),VERTCS(2,11),VER(2,10)
  DIMENSION ACARD(20)
  DATA BLANK/,/,COMMA/,/,OP/,/,CP/,/,
  *AST/,/,ENDBCD/,/SEND/,/
  DATA C/,/CLAS/,/S/,/SUBC/,/
  DATA D/,/DEST/,/
  DATA T/,/TYPE/,/
  DO 50 I=1,2
  DO 50 J=1,10
  50 VER(I,J)=0
C SET UP REREAD BUFFER
  PRUNIT=30
  CALL RFREAD(RRUNIT,80)
C NOW PUT THE CARD INTO THE BUFFER
  1 READ(21,100)(ACARD(I),I=1,20)
  100 FORMAT(20A4)
  WRITE(RRUNIT,100)(ACARD(I),I=1,20)
  REWIND RRUNIT
C READ IN FIRST CARD
  READ(RRUNIT,20) FLDNAM
  20 FORMAT(A4)
  REWIND RRUNIT
  IF (FLDNAM.NE.T) GO TO 55
  LAREAD = -4
  RETURN
  55 CONTINUE
  IF(FLDNAM.NE.D)GO TO 19
  LAREAD=-3
  RETURN
  19 CONTINUE
  IF(FLDNAM.NE.C) GO TO 24
  LAREAD=-1
  RETURN
  24 IF(FLDNAM.NE.S) GO TO 21
  LAREAD=-2
  RETURN
  21 IF(FLDNAM.NE.ENDBCD) GO TO 22
  LAREAD=0
  RETURN
C REREAD FIRST CARD
  22 READ(RRUNIT,23) CARD
  23 FORMAT(10X,62A1)
  REWIND RRUNIT
  COL=0
  II=0
  NC=0
C FIND (
  11 J=NXTCHR(CARD,COL)
  IF(J.EQ.BLANK) GO TO 1
  IF(J.EQ.AST) GO TO 10
  IF(J.NE.OP) GO TO 3
  9 I=0
  K=0
  KK=0
  NUM=0
  4 COL=COL+1
  IF(COL.GT.62) GO TO 35
  IF(CARD(COL).EQ.BLANK) GO TO 4
  IF((I.EQ.0).AND.(CARD(COL).EQ.COMMA)) GO TO 3
  IF((I.EQ.1).AND.(CARD(COL).EQ.COMMA)) GO TO 7
  IF((KK.EQ.0).AND.(CARD(COL).EQ.CP)) GO TO 3
  IF((KK.EQ.1).AND.(CARD(COL).EQ.CP)) GO TO 8
  CALL I4A1B(CARD(COL),I,NW)
  NUM=10*NUM+NW
  IF((NW.LT.0).OR.(NW.GT.9)) GO TO 3
  I=1
  IF(K.EQ.1) KK=1
  GO TO 4
  7 IF(II.EQ.0) GO TO 30
C VERTEX SAMPLE NUMBER
  NC=NC+1
  VER(1,NC)=NUM
  K=1
  NUM=0
  GO TO 4
  8 IF(II.EQ.0) GO TO 31
C VERTEX LINE NUMBER

```

LAR00010
 LAR00020
 LAR00030
 LAR00040
 LAR00050
 LAR00060
 LAR00070
 LAR00080
 LAR00090
 LAR00100
 LAR00110
 LAR00120
 LAR00130
 LAR00140
 LAR00150
 LAR00160
 LAR00170
 LAR00180
 LAR00190
 LAR00200
 LAR00210
 LAR00220
 LAR00230
 LAR00240
 LAR00250
 LAR00260
 LAR00270
 LAR00280
 LAR00290
 LAR00300
 LAR00310
 LAR00320
 LAR00330
 LAR00340
 LAR00350
 LAR00360
 LAR00370
 LAR00380
 LAR00390
 LAR00400
 LAR00410
 LAR00420
 LAR00430
 LAR00440
 LAR00450
 LAR00460
 LAR00470
 LAR00480
 LAR00490
 LAR00500
 LAR00510
 LAR00520
 LAR00530
 LAR00540
 LAR00550
 LAR00560
 LAR00570
 LAR00580
 LAR00590
 LAR00600
 LAR00610
 LAR00620
 LAR00630
 LAR00640
 LAR00650
 LAR00660
 LAR00670
 LAR00680
 LAR00690
 LAR00700
 LAR00710
 LAR00720
 LAR00730
 LAR00740
 LAR00750
 LAR00760
 LAR00770
 LAR00780
 LAR00790

ORIGINAL PAGE IS
OF POOR QUALITY

FILE: LAREAD

```
VER(2,NC)=NUM
C CHECK FOR COMMA OR ASTERISK
6 J=NXTCHR(CARD,COL)
IF(J.EQ.BLANK) GO TO 2
IF(J.EQ.AST) GO TO 10
IF(J.NE.COMMA) GO TO 3
GO TO 11
3 WRITE(6,13) CARD
13 FORMAT(' ERROR IN FIELD CARD TERMINATING RUN'/10X,62A1)
CALL CMERR
5 WRITE(6,15) CARD
15 FORMAT(10X,62A1/' INCORRECT FIELD CARD,TERMINATING RUN')
CALL CMERR
C DETERMINE RECTANGULAR FIELD COORDINATES
2 IF((NC.LT.1).OR.(NC.GT.10)) GO TO 3
NT1=0
NT3=0
NT2=1000000
NT4=1000000
DO 14 N=1,NC
IF((VER(1,N).EQ.0).OR.(VER(2,N).EQ.0)) GO TO 5
IF(VER(1,N).GT.NT1) NT1=VER(1,N)
IF(VER(2,N).LT.NT4) NT4=VER(2,N)
IF(VER(2,N).GT.NT3) NT3=VER(2,N)
IF(VER(1,N).GE.NT2) GO TO 14
NT2=VER(1,N)
CNT=N
14 CONTINUE
FLDINF(1)=NT4
FLDINF(2)=NT3
FLDINF(4)=NT2
FLDINF(5)=NT1
C SFT UP VERTICES IN CLOCKWISE ORDER WITH SMALLEST SAMPLE FIRST
DO 32 I=1,NC
IF(CNT.GT.NC) CNT=1
VERTCS(1,I)=VER(1,CNT)
VERTCS(2,I)=VER(2,CNT)
32 CNT=CNT+1
VERTCS(1,NC+1)=VERTCS(1,1)
VERTCS(2,NC+1)=VERTCS(2,1)
35 IAREAD=1
NC=NC+1
RETURN
30 FLDINF(6)=NUM
K=1
NUM=0
GO TO 4
31 FLDINF(3)=NUM
II=1
GO TO 6
C READ CONTINUATION CARD
10 READ(21,23) CARD
COL=0
GO TO 11
END
```

LAR00800
LAR00810
LAR00820
LAR00830
LAR00840
LAR00850
LAR00860
LAR00870
LAR00880
LAR00890
LAR00900
LAR00910
LAR00920
LAR00930
LAR00940
LAR00950
LAR00960
LAR00970
LAR00980
LAR00990
LAR01000
LAR01010
LAR01020
LAR01030
LAR01040
LAR01050
LAR01060
LAR01070
LAR01080
LAR01090
LAR01100
LAR01110
LAR01120
LAR01130
LAR01140
LAR01150
LAR01160
LAR01170
LAR01180
LAR01190
LAR01200
LAR01210
LAR01220
LAR01230
LAR01240
LAR01250
LAR01260
LAR01270
LAR01280
LAR01290
LAR01300
LAR01310
LAR01320
LAR01330
LAR01340

FILE LINERO

```

SUBROUTINE LINERO(/IDATA/,ENDTAP)
IMPLICIT INTEGER (A-Z)

C*
C*
C* ENTRY FOR READING AND UNPACKING ONE SCAN LINE OF DATA
C*
LOGICAL*1 ISCAN(4),BYTE(4),IDATA(1),IBUF(13500),IZERO(4)
C INCLUDE CMHK17
C*START
COMMON /HUFF/IBUFF(3375)
COMMON /TAPERD/ IUNIT,(FRST,FSCAN,SAMEND,SAMINC,READY,NSCAN,
* LINC,ID(200),DSL,LBUF(30),JREC(30),IHYTE(30),NHUFS,FILENO,LINEND
* LININC,NSAMP,NOCHAN,FORMT
C*END
EQUIVALENCE (SCAN,ISCAN(1)),(IZERO(1),ZERO)
EQUIVALENCE (IHUFF,IBUF(1))
EQUIVALENCE (ID(1),NRPDS),(ID(2),NCPR),
* (ID(3),NPRC),(ID(4),ANCLNG),
* (ID(5),NC),(ID(6),NS),
* (ID(7),NHITS),(ID(8),DOI),
* (ID(9),NDSPR),(ID(10),NCAR),
* (ID(11),SVD),(ID(16),PRSZ)
C ZERO OUT IDATA
C
SCAN = 0
TOTPIX=NSAMP*NOCHAN*4
ZERO=0
DO 180 J=1,TOTPIX
180 IDATA(J)=ZERO(4)
IF (READY)190,190,200
190 WRITE (6,410)
C*
200 IADR = 4
MAXREC = PRSZ
RUF = 1
REC = 0
IF (FORMT.EQ.4)GO TO 2000
IF (NDSPR.EQ.1)GO TO 195
IF (FSCAN.EQ.NSCAN)GO TO 195
IF ((FSCAN+NDSPR-1).LE.NSCAN)GO TO 196
195 CALL HUFFILL(REC,IUNIT,MAXREC,IBUF,NRPDS,ENDTAP,IERR)
IF (FORMT.EQ.3)GO TO 1000
IF (FORMT.EQ.1) ISCAN(3) = IBUF(71)
IF (FORMT.EQ.1) ISCAN(4) = IBUF(72)
IF (FORMT.EQ.2) ISCAN(3) = IBUF(1)
IF (FORMT.EQ.2) ISCAN(4) = IBUF(2)
C
IF (SCAN.EQ.FSCAN) GO TO 196
C
CALL SEARCH(&250,&235,ENDTAP,IBUF,NRPDS,NDSPR)
MAXREC=PRSZ
RUF = 1
REC = 0
CALL HUFFILL(REC,IUNIT,MAXREC,IBUF,NRPDS,ENDTAP,IERR)
196 CONTINUE
ADD = (NSCAN-FSCAN)*DSL
DO 230 IFT=1,NOCHAN
201 IF (LBUF(IFT).EQ.HUFF)GO TO 205
CALL HUFFILL(REC,IUNIT,MAXREC,IBUF,NRPDS,ENDTAP,IERR)
RUF=RUF+1
GO TO 201
205 CONTINUE
J=JREC(IFT)
JJ=(J-1)*MAXREC
C*
C* CHECK STATUS OF THIS RECORD BEFORE UNPACKING
C*
IF (ENDTAP.EQ.-1) GO TO 250
C*
C* UNPACK DATA FOR THIS FEATURE
C*
220 IP = ADD + IHYTE(IFT)+JJ
DO 225 II=1,NSAMP
IDATA(IADR+4*(II-1)) = IBUF(IP+SAMINC*(II-1))
225 CONTINUE
IADR = IADR + NSAMP*4

```

LIN00010
 LIN00020
 LIN00030
 LIN00040
 LIN00050
 LIN00060
 LIN00070
 LIN00080
 LIN00090
 LIN00100
 LIN00110
 LIN00120
 LIN00130
 LIN00140
 LIN00150
 LIN00160
 LIN00170
 LIN00180
 LIN00190
 LIN00200
 LIN00210
 LIN00220
 LIN00230
 LIN00240
 LIN00250
 LIN00260
 LIN00270
 LIN00280
 LIN00290
 LIN00300
 LIN00310
 LIN00320
 LIN00330
 LIN00340
 LIN00350
 LIN00360
 LIN00370
 LIN00380
 LIN00390
 LIN00400
 LIN00410
 LIN00420
 LIN00430
 LIN00440
 LIN00450
 LIN00460
 LIN00470
 LIN00480
 LIN00490
 LIN00500
 LIN00510
 LIN00520
 LIN00530
 LIN00540
 LIN00550
 LIN00560
 LIN00570
 LIN00580
 LIN00590
 LIN00600
 LIN00610
 LIN00620
 LIN00630
 LIN00640
 LIN00650
 LIN00660
 LIN00670
 LIN00680
 LIN00690
 LIN00700
 LIN00710
 LIN00720
 LIN00730
 LIN00740
 LIN00750
 LIN00760

ORIGINAL PAGE IS
OF POOR QUALITY

FILE LINERD.

| | | |
|----|--|----------|
| C* | 230 CONTINUE | LIN00770 |
| C* | FINISHED UNPACKING ONE SCAN LINE OF DATA | LIN00780 |
| C* | IF ((NSCAN+LININC).GT.LINEND) GO TO 260 | LIN00790 |
| C* | MAKE SURE ALL BUFFERS FOR THIS DATA SET HAVE BEEN READ | LIN00800 |
| C* | 231 IF (RUF.EQ.NRUF) GO TO 235 | LIN00810 |
| | CALL HUFILL(REC,IUNIT,MAXREC,IBUF,NRPDS,ENDTAP,IERR) | LIN00820 |
| | IF (ENDTAP.EQ.-1) GO TO 250 | LIN00830 |
| | RUF=RUF+1 | LIN00840 |
| | GO TO 231 | LIN00850 |
| | 235 CONTINUE | LIN00860 |
| | NSCAN=NSCAN+LININC | LIN00870 |
| | IF (NSCAN.LT.(FSCAN+NDSPR)) RETURN | LIN00880 |
| | IF (NSCAN.GT.(FSCAN+NDSPR)) GO TO 236 | LIN00890 |
| | FSCAN = FSCAN + NDSPR*(1 + LINC/NRPOS) | LIN00900 |
| | RETURN | LIN00910 |
| | 236 FSCAN=FSCAN + NDSPR*(1 + LINC/NRPOS) | LIN00920 |
| | DO 237 II=1,LINC | LIN00930 |
| | 237 READ(IUNIT,420)DUMMY | LIN00940 |
| | IF (NSCAN.LT.(FSCAN+NDSPR)) GO TO 240 | LIN00950 |
| | READ(IUNIT,420)DUMMY | LIN00960 |
| | FSCAN=FSCAN+NDSPR | LIN00970 |
| | 240 CONTINUE | LIN00980 |
| | RETURN | LIN00990 |
| | 250 IF (NSCAN.GT.LINSTR) NSCAN=NSCAN-LININC | LIN01000 |
| | WRITE (6,320) NSCAN | LIN01010 |
| | IF (FILENO.EQ.0) GO TO 255 | LIN01020 |
| C* | RACK SPACE 1 FILE AND POSITION AT FIRST SCAN LINE | LIN01030 |
| C* | RSKIP = (NSCAN-IFRST) * NDSPR + 1 | LIN01040 |
| | WRITE(6,560)RSKIP | LIN01050 |
| | 560 FORMAT(' RACKSPACE',I3) | LIN01060 |
| | DO 253 II=1,RSKIP | LIN01070 |
| | 253 RACKSPACE IUNIT | LIN01080 |
| | GO TO 257 | LIN01090 |
| C* | REWIND TAPE AND POSITION AT FIRST SCAN LINE | LIN01100 |
| C* | 255 REWIND IUNIT | LIN01110 |
| | READ(IUNIT,420)DUMMY | LIN01120 |
| | 257 FSCAN = IFRST | LIN01130 |
| | 260 READY = -1 | LIN01140 |
| | RETURN | LIN01150 |
| C | UNPACK SCAN LINE OF DATA FOR LANDSAT 1 OR 2 | LIN01160 |
| C | 1000 SAMSTR=JREC(1) | LIN01170 |
| | DO 1100 I=1,NOCHAN | LIN01180 |
| | IJ=-1 | LIN01190 |
| | DO 1200 II=SAMSTR,SAMEND,SAMINC | LIN01200 |
| | IJ=IJ+1 | LIN01210 |
| | JJ=II | LIN01220 |
| | KK=1 | LIN01230 |
| | IF (MOD(II,2).EQ.0) JJ=JJ-1 | LIN01240 |
| | IF (MOD(II,2).EQ.0) KK=1 | LIN01250 |
| | IADD=I*BYTE(1)+(JJ-1)*4+KK | LIN01260 |
| | DATA(IADR+4*IJ)=IBUF(IADD) | LIN01270 |
| | 1200 CONTINUE | LIN01280 |
| | IADR=IADR+NSAMP*4 | LIN01290 |
| | 1100 CONTINUE | LIN01300 |
| | GO TO 235 | LIN01310 |
| C | UNPACK SCAN LINE FOR LANDSAT III | LIN01320 |
| C | 2000 J=1 | LIN01330 |
| | SAMSTR=IBUF(1) | LIN01340 |
| | DO 2500 I=1,NOCHAN | LIN01350 |
| | IF (I*BYTE(1).EQ.J) GO TO 2200 | LIN01360 |
| | LIM=I*BYTE(1)-J | LIN01370 |
| | DO 2100 II=1,LIM | LIN01380 |
| | READ(IUNIT,420)DUMMY | LIN01390 |
| | 2100 CONTINUE | LIN01400 |
| | 2200 CALL HUFILL(REC,IUNIT,MAXREC,IBUF,1,ENDTAP,IERR) | LIN01410 |
| | DO 2300 II=1,NSAMP | LIN01420 |
| | | LIN01430 |
| | | LIN01440 |
| | | LIN01450 |
| | | LIN01460 |
| | | LIN01470 |
| | | LIN01480 |
| | | LIN01490 |
| | | LIN01500 |
| | | LIN01510 |
| | | LIN01520 |

FILE LINERO

| | | | | |
|------|---|---|----|-----|
| 2300 | DATA(IADR+4*(II-1))=IHUF(13+SAMINC*(II-1)+SAMSTR) | L | NO | 530 |
| | CONTINUE | L | NO | 540 |
| | IADR=IADR+4*NSAMP | L | NO | 550 |
| | J=IAYE(I)+1 | L | NO | 560 |
| 2400 | CONTINUE | L | NO | 570 |
| | IF(J.GT.NRPDS)GO TO 235 | L | NO | 580 |
| | LIM=NRPDS-J+1 | L | NO | 590 |
| | DO 2600 II=1,LIM | L | NO | 600 |
| | READ(IUNIT,420)DUMMY | L | NO | 610 |
| 2500 | CONTINUE | L | NO | 620 |
| | GO TO 235 | L | NO | 630 |
| 320 | FORMAT(' FIELD BOUNDARY FOR THIS FIELD DEFINED BEYOND SCOPE OF IADR | L | NO | 640 |
| | *TA// THIS FLIGHT LINE CONTAINS '.16.' SCAN LINES') | L | NO | 650 |
| 410 | FORMAT(' FLDINT MUST BE CALLED TO INITIALIZE PARAMETERS FOR A NEW | L | NO | 660 |
| | *FIELD') | L | NO | 670 |
| 420 | FORMAT(1A4) | L | NO | 680 |
| | END | L | NO | 690 |

FILE: LISTLC

```

C      FIELDS - CATEGORY NAME AND DOT TYPE FOR DOT 1 STORED IN      S00010
C      FIELD(1,1) AND FIELD(4,1)                                     S00020
C      STAMNT - INITIALLY SET TO 1, SWITCHED TO INDICATE DOTS BEING   S00030
C      TAKEN FROM CURRENTLY READ CARD.                                S00040
C      IPT - INITIALLY SET TO 1, INDEX NUMBER FOR FIELD VERTEX INFORMATION S00050
C      VERTEX - VERTEX INFORMATION FOR EACH DOT.                       S00060
C      SUBROUTINE LISTLC(FIELDS,STAMNT,*,*,SWCHG,INIT,IUNIT,IFILE,IPT, S00070
C      *VERTEX)                                                       S00080
C      IMPLICIT INTEGER (A-Z)                                         S00090
C      LOGICAL*1 LCARD(300),LCATNM(4)                                S00100
C      REAL DUM                                                        S00110
C      DIMENSION FIELDS(4,1),VERTEX(1),CARD(75),NDOTS(30)           S00120
C      DIMENSION ACARD(80)                                             S00130
C      LOGICAL SWITCH                                                  S00140
C      DATA SWITCH/.TRUE./,ENDRCD/'SEN'/,                             S00150
C      *CATNM/' ' /                                                    S00160
C      INCLUDE CMHK14                                                  S00170
C      INCLUDE COMAK1                                                  S00180
C      COMMON/INFORM/NOCLS2,NOSUB2,NOFET2,VARS22,TOTVT2,NOFLD2,       S00190
C      *      AVAR2,COVAR2,CLS1D2,SUHN02,SUHDS2,FLDSV2,VERTX2,       S00200
C      *      FFTVC2(30),SU=VC2(75),SUHPT(75),CLSV2(60),            S00210
C      *      KFPPTS(60),NOGWP,GWPNAM(60),GRPDEX(61),                S00220
C      *      GRPCHK(61),GROUPS(124)                                  S00230
C      *      COMMON /DOTVEC/TYPE,CATNM(60),NOCAT,TOTVEC,FLDINF(6),PRTKEY S00240
C      *      ,SIZE,LACIE                                             S00250
C      *      DIMENSION IBUF(80)                                       S00260
CSEND      EQUIVALENCE (LCATNM(1),CATNM),(CARD(1),LCARD(1))          S00270
C      IF (INIT.NE.0) GO TO 5                                          S00280
C      REWIND IUNIT                                                    S00290
C      CALL FCFMFL(IUNIT,IFILE,ISTAT)                                  S00300
C      READ (IUNIT,1010) (IBUF(I), I=1,80)                            S00310
C      1010  FORMAT(M0A1)                                              S00320
C      WRITE(6,1020) (IBUF(I), I=1,80)                                S00330
C      1020  FORMAT(1H0,80A1)                                          S00340
C      5      INIT = 1                                                 S00350
C      IF (STAMNT.EQ.2) GO TO 30                                       S00360
C      IF (.NOT.SWITCH) GO TO 20                                       S00370
C      CALL REREAD(30,80)                                              S00380
C      10  READ(IUNIT,103) (ACARD(I), I=1,80)                          S00390
C      103  FORMAT(M0A1)                                              S00400
C      WRITE(30,103) (ACARD(I), I=1,80)                                S00410
C      REWIND 30                                                       S00420
C      READ(30,1000) ID,TYPE,CARD                                     S00430
C      REWIND 30                                                       S00440
C      1000 FORMAT(A3,1X,11,75A1)                                       S00450
C      IF (TYPE.EQ.0) GO TO 20                                         S00460
C      IF (SWCHG.NE.0) GO TO 40                                         S00470
C      TYPE = TYPES                                                    S00480
C      READ CARD                                                       S00490
C      COL = 0                                                         S00500
C      CATNM = NATCHP(CARD,COL)                                         S00510
C      IF NEXT CHAR IS NOT A CAT. NAME, CORRECT COL COUNT TO READ NUM S00520
C      IF (CATNM.GT.0) GO TO 21                                         S00530
C      LINDEX=4*COL+1                                                  S00540
C      LCATNM(2)=LCARD(LINDEX)                                         S00550
C      COL=COL+1                                                       S00560
C      IF (CATNM.EQ.CATNM) GO TO 23                                     S00570
C      NOCAT=NOCAT + 1                                                 S00580
C      CATNM(NOCAT)=CATNM                                              S00590
C      CATNM = CATNM                                                    S00600
C      GO TO 23                                                         S00610
C      21  COL=COL - 1                                                  S00620
C      23  NOCARD=0                                                    S00630
C      CALL NUMBP(NDOTS,NOCARD,CARD,COL)                               S00640
C      IF (NOCARD.EQ.0) GO TO 10                                       S00650
C      ICNT = 0                                                         S00660
C      STAMNT = 2                                                       S00670
C      SWITCH = .TRUE.                                                 S00680
C      GO TO 100                                                        S00690
C      TEST FOR END OF DOTS TO BE PROCESSED ON CARD                  S00700
C      30  IF (ICNT.LT.NOCARD) GO TO 100                               S00710
C      READ NEXT CARD                                                  S00720
C      S00730
C      S00740
C      S00750
C      S00760
C      S00770
C      S00780
C      S00790

```

FILE: LISTLC

```

C
  STAMNT = 1
  ICNT = 0
  READ(IUNIT,103)(ACARD(I),I=1,40)
  WRITE(30,103)(ACARD(I),I=1,80)
  REWIND 30
  READ(30,1000)ID,TYPE,CARD
  REWIND 30
  IF(ID.EQ.FNDHCD)RETURN 3
  IF(TYPE.EQ.TYPES)GO TO 20
  SWITCH = .FALSE.
  SWCHG = SWCHG + 1
  IF(SWCHG.GT.1)GO TO 40
  TYPE = TYPES
  IPT = 0
C***** CHANGED JUNE 28 1978
  RETURN 2
C
C 100 ICNT = ICNT + 1
  NOFLD2 = NOFLD2 + 1
C
C   COMPUTE LINE INCREMENT
  NN = NDOTS(ICNT)
  NI = IARS(NN) / 100000000
  LI = IARS(NN) - NI * 100000000
  IF(LI.GE.100000000)NI = NI + 1
C
C   COMPUTE SAMPLE INCREMENT
  KK=1
  IF(NN.LT.0)KK=-1
  LI = NI * KK
  N2 = NN - LI * 100000000
  N3 = IARS(N2)/10000
  SI = IARS(N2)-N3 * 10000
  IF(SI.GE.10000)N3 = N3 + 1
  KK=1
  IF(N2.LT.0)KK=-1
  SI = N3 * KK
  LACI = N2 - SI * 10000
  LU = (LACI-1)/19
  LR = (LR+1) * 10
  LS = LR - 1
  LS = LS / 10
  LS = 10 * (LACI - (LS*19))
  L = LR - LI
  S = LS + SI
C
C   STORE DOT INFO
  FIELDS(1,NOFLD2) = CATNM
  FIELDS(4,NOFLD2) = 2
  FLDINF(1) = L
  FLDINF(2) = L
  FLDINF(3) = 1
  FLDINF(4) = S
  FLDINF(5) = S
  FLDINF(6) = 1
  IF(IPT.NE.0)GO TO 35
  IPT = -3
  IPT = IPT + 4
  VERTEX(IPT) = S
  VERTEX(IPT+1)=L
  VERTEX(IPT+2)=S
  VERTEX(IPT+3)=L
  RETURN 1
  40 WRITE(5,2000)
  2000 FORMAT(//EX. ERROR HAS OCCURRED IN READING LACIE FORMATTED DOT CAR
  DS = SUBROUTINE FLULAC - EXIT TAKEN)
  RETURN 3
  END

```

L 500800
 L 500810
 L 500820
 L 500830
 L 500840
 L 500850
 L 500860
 L 500870
 L 500880
 L 500890
 L 500900
 L 500910
 L 500920
 L 500930
 L 500940
 L 500950
 L 500960
 L 500970
 L 500980
 L 500990
 L 501000
 L 501010
 L 501020
 L 501030
 L 501040
 L 501050
 L 501060
 L 501070
 L 501080
 L 501090
 L 501100
 L 501110
 L 501120
 L 501130
 L 501140
 L 501150
 L 501160
 L 501170
 L 501180
 L 501190
 L 501200
 L 501210
 L 501220
 L 501230
 L 501240
 L 501250
 L 501260
 L 501270
 L 501280
 L 501290
 L 501300
 L 501310
 L 501320
 L 501330
 L 501340
 L 501350
 L 501360
 L 501370
 L 501380
 L 501390
 L 501400
 L 501410
 L 501420
 L 501430
 L 501440
 L 501450
 L 501460
 L 501470
 L 501480
 L 501490
 L 501500
 L 501510
 L 501520

FILE: MATVEC

```
      SUBROUTINE MATVEC(A,H,C,L,M)
C  MULTIPLY MATRIX A BY VECTOR B AND STORE IN VECTOR C
      DIMENSION A(L,M),B(M),C(L)
      DO 10 I=1,L
      SUM=0.
      DO 5 K=1,M
      SUM=SUM+A(I,K)*B(K)
10    C(I)=SUM
      RETURN
      END
```

ORIGINAL PAGE IS
OF POOR QUALITY

FILE: MTMDAT

```

C      SUBROUTINE MTMDAT(A,B,C,L,M,N,D,DD)
C      MULTIPLY MATRIX A BY THE TRANSPOSE OF B AND STORE IN DD
C      A LOWER TRIANGULAR MATRIX
      DIMENSION A(L,M),B(N,M),C(L,N),D(L),DD(1)
      DO 60 I=1,L
      DO 60 J=1,N
      SUM=0.0
      DO 55 K=1,M
55  SUM=SUM+A(I,K)*B(J,K)
      C(I,J)=SUM
      IF(I.EQ.J) D(I)=SORT(SUM)
60  CONTINUE
      MM=0
      KK=0
      DO 1 II=1,L
      KK=KK+1
      DO 1 LL=1,KK
      MM=MM+1
      DD(MM)=C(II,LL)
1  CONTINUE
      RETURN
      END
```

FILE: MTMLS6

| | | |
|----|--|----------|
| | SUBROUTINE MTMLS6(A,B,C,M,N) | MTM00010 |
| C | MULTIPLY MATRIX A BY B AND STORE IN C. B IS STORED IN SYMMETRIC NOTATION | MTM00020 |
| C | DIMENSION A(M,N),B(1),C(M,N) | MTM00030 |
| | DO 50 J=1,M | MTM00040 |
| | LE=0 | MTM00050 |
| | DO 40 I=1,N | MTM00060 |
| | LB=LE+1 | MTM00070 |
| | LE=LE+1 | MTM00080 |
| | SUM=0. | MTM00090 |
| | K=0 | MTM00100 |
| | DO 35 L=LB,LE | MTM00110 |
| | K=K+1 | MTM00120 |
| 35 | SUM=SUM+A(J,K)*B(L) | MTM00130 |
| | IF(I.EQ.N) GO TO 40 | MTM00140 |
| | KS=K+1 | MTM00150 |
| | L=LE | MTM00160 |
| | DO 36 K=KS,N | MTM00170 |
| | L=L+I+K-KS | MTM00180 |
| 36 | SUM=SUM+A(J,K)*B(L) | MTM00190 |
| 40 | C(J,I)=SUM | MTM00200 |
| 50 | CONTINUE | MTM00210 |
| | RETURN | MTM00220 |
| | END | MTM00230 |
| | | MTM00240 |
| | | MTM00250 |

FILE: NAMSTA

| | | |
|----|--|----------|
| C | NAMSTA ASSIGNS NAMES TO CLUSTERS AND UPDATES STAT INFO | NAM00010 |
| C | SUBROUTINE NAMSTA(SURNAM,CATVEC,SUBNO,NOSUB2,CATNAM,NOCAT) | NAM00020 |
| C | IMPLICIT INTEGER (A-Z) | NAM00030 |
| | DIMENSION SURNAM(60),CATNAM(60) | NAM00040 |
| | DIMENSION CATVEC(60),SUBNO(1) | NAM00050 |
| | INTEGER*4 I4(3) | NAM00060 |
| | LOGICAL*1 L1(12) | NAM00070 |
| | EQUIVALENCE(L1(1),I4(1)) | NAM00080 |
| | K = 0 | NAM00090 |
| C | ASSIGN NAMES TO CLUSTERS | NAM00100 |
| C | DO 20 I=1,NOCAT | NAM00110 |
| | L = 0 | NAM00120 |
| | DO 20 J=1,NOSUB2 | NAM00130 |
| | IF (CATVEC(J) .NE. I) GO TO 20 | NAM00140 |
| | K = K + 1 | NAM00150 |
| | L = L + 1 | NAM00160 |
| C | USE FIRST 2 CHAR OF CATEGORY NAME + 2 DIGITS | NAM00170 |
| C | I4(1) = CATNAM(I) | NAM00180 |
| | CALL RN141(I4(2),2,L) | NAM00190 |
| | L1(3)=L1(5) | NAM00200 |
| | L1(4)=L1(9) | NAM00210 |
| C | SURNAM(K) = I4(1) | NAM00220 |
| C | 20 CONTINUE | NAM00230 |
| C | CHECK FOR NULL CATEGORY | NAM00240 |
| | I=NOCAT | NAM00250 |
| 30 | IF (SURNAM(I) .NE. 0) GO TO 60 | NAM00260 |
| | IF (I .EQ. NOCAT) GO TO 55 | NAM00270 |
| | DO 50 J=I,NOCAT | NAM00280 |
| | CATNAM(J) = CATNAM(J+1) | NAM00290 |
| 50 | SURNAM(J) = SURNAM(J+1) | NAM00300 |
| 55 | NOCAT = NOCAT - 1 | NAM00310 |
| 60 | I=I-1 | NAM00320 |
| | IF (I.GT.0) GOTO 30 | NAM00330 |
| | RETURN | NAM00340 |
| C | END | NAM00350 |
| | | NAM00360 |
| | | NAM00370 |
| | | NAM00380 |
| | | NAM00390 |
| | | NAM00400 |
| | | NAM00410 |
| | | NAM00420 |
| | | NAM00430 |
| | | NAM00440 |
| | | NAM00450 |
| | | NAM00460 |
| | | NAM00470 |
| | | NAM00480 |

FILE: NUMBER

```
FUNCTION NUMBER(CARD,COL,NUMVEC,NOW)
IMPLICIT INTEGER (A-Z)
DIMENSION CARD(1),NUMVEC(1)
DATA CRDSIZ/62/,VECSIZ/100/,BLANK/' ',COMMA/',',/
DATA ZERO/'0',NINE/'9'/
NEXT = NOW + 1
IF (NEXT .LE. 0 .OR. NEXT .GT. VECSIZ) NEXT = 1
J = 0
L = COL+1
IF (L .GT. CRDSIZ) GO TO 92
VK=VECSIZ
DO 80 J=NEXT,VK+1
JJ = J
NUM = 0
ITRIG=0
DO 60 COL=L,CRDSIZ+1
IF (CARD(COL).EQ.BLANK) GO TO 60
IF (CARD(COL).EQ.COMMA) GO TO 70
IF (CARD(COL).LT.ZERO.OR.CARD(COL).GT.NINE) GO TO 90
CALL I4A1HN(CARD(COL),1,NWORD)
NUM = 10 * NUM + NWORD
ITRIG=1
60 CONTINUE
COL = CRDSIZ
GO TO 90
70 NUMVEC(J) = NUM
L = COL+1
IF (L .GT. CRDSIZ) GO TO 92
80 CONTINUE
J = VECSIZ
90 IF (ITRIG.EQ.1) GO TO 91
J = J - 1
GO TO 92
91 NUMVEC(J) = NUM
92 NUMBER = J
C 106 WRITE( 6,106) (CARD(K),K=1,62),COL,NUMBER,(NUMVEC(K),K=1,J)
C 106 FORMAT(' NUMBER ENTERED',/,',62A1,I10/,',',15,18I3)
RETURN
C*****
C***** FUNCTION ENTRIES MUST RETURN VALUE IN ORIGINAL FUNCTION NAME
C*****
END
```

ORIGINAL PAGE IS
OF POOR QUALITY

FILE: NUMBR

| | | |
|-----|---|----------|
| C* | SUBROUTINE NUMBR WILL PROCESS ONE CARD AT A TIME. | NUM00010 |
| C* | IT READS AND STORES ALL NUMBERS IN ARRAY NDOTS WITH | NUM00020 |
| C* | NDCARD AS AN INDEX. BLANKS ARE THE ONLY RECONIZED | NUM00030 |
| C* | DELIMITERS. | NUM00040 |
| | SUBROUTINE NUMBR(NDOTS,NDCARD,CARD,COL) | NUM00050 |
| | IMPLICIT INTEGER (A-Z) | NUM00060 |
| | DIMENSION NDOTS(1),CARD(1) | NUM00070 |
| | DATA BLANK/' ' /,CRDSIZ/75/ | NUM00080 |
| | NUM=0 | NUM00090 |
| | NC = COL + 1 | NUM00100 |
| 5 | IF (NC.GT.CRDSIZ)GO TO 50 | NUM00110 |
| | DO 10 I=NC,CRDSIZ | NUM00120 |
| | IF (CARD(I).EQ.BLANK)GO TO 7 | NUM00130 |
| | CALL I4A18V(CARD(I),1,NWORD) | NUM00140 |
| | NUM = NUM*10 + NWORD | NUM00150 |
| | GO TO 30 | NUM00160 |
| 7 | IF (NUM.LT.1)GO TO 30 | NUM00170 |
| | IF (NUM.GT.209)WRITE (6,500)NUM | NUM00180 |
| | NDCARD=NDCARD + 1 | NUM00190 |
| | NDOTS(NDCARD)=NUM | NUM00200 |
| | NUM = 0 | NUM00210 |
| 30 | CONTINUE | NUM00220 |
| 10 | CONTINUE | NUM00230 |
| 500 | FORMAT(7/5X,'LACIE DOT READ THAT IS GREATER THAN SIZE LIMIT | NUM00240 |
| | :OF 209 - EXECUTION CONTINUED WITH VALUE READ OF '.14) | NUM00250 |
| 50 | CONTINUE | NUM00260 |
| | RETURN | NUM00270 |
| | END | NUM00280 |

U

CALL J=NXTCHR(CARD,COL)

PURPOSE LOCATES THE NEXT NON BLANK SYMBOL IN 'CARD'

IMPLICIT INTEGER (A-Z)

DIMENSION CARD(1)

DATA CROSIZ/62/,BLANK/' '

DATA CRDS
L = COL+1

```
IF (L.GT.CROSSIZ) GO TO 40
```

00 30 COL=L.CARDSIZ

```
NXTCHR = CARD(COL)
```

IF (NXTCHR.NE.BLANK) GO TO 50

30 CONTINUE

COL = CRD\$12-1

```

40      NEXTCHR = BLANK

```

50 CONTINUE

```
C      WRITE (6,104) (CAPD(K),K=1,62),COL,NXTCHR
```

```
104 FORMAT(' NATCHR ENTERED'// ' ',62A1,110// ' ',A4)
```

RETURN

END

```
NXT00010
NXT00020
NXT00030
NXT00040
NXT00050
NXT00060
NXT00070
NXT00080
NXT00090
NXT00100
NXT00110
NXT00120
NXT00130
NXT00140
NXT00150
NXT00160
NXT00170
NXT00180
NXT00190
NXT00200
NXT00210
NXT00220
NXT00230
NXT00240
NXT00250
NXT00260
NXT00270
NXT00280
NXT00290
NXT00300
```

~~19-59~~
432

FILE: ORDER

```
SUBROUTINE ORDER(VEC,N)
  IMPLICIT INTEGER(A-Z)
  DIMENSION VEC(1)
  LOGICAL SWITCH
  IF(N.LE.1)RETURN
  M=N-1
  5 SWITCH=.FALSE.
  DO 10 I=1,M
    IF(VEC(I).LE.VEC(I+1))GO TO 10
    TEMP=VEC(I)
    VEC(I)=VEC(I+1)
    VEC(I+1)=TEMP
    SWITCH=.TRUE.
  10 CONTINUE
  IF(SWITCH)GO TO 5
  RETURN
END
```


FILE: PRINT

```
SUBROUTINE PRINT(KKT,IPLACE,MEANS,STDEV,CLD,FLDINF,N)
C      IMPLICIT INTEGER(A-X)
C      INCLUDE COMPK5.LIST
C      INCLUDE COMPK6.LIST
COMMON/PASS/STOP,LNCAT,NMIN,KRN,STDMAX,DLMIN,SEP,
*      MAP,SPTRIG,IRD,KPTS,NOPTS,PUNCH,
*      ICHN,CHNTHS,ICHAIN(42),NWDS,IREGIN,BEGIN1,
*      HEGIN2,HEGIN3,CLSNAM,NOFLD,IPT,TOTWRD,TOTPTS,
*      NCLASS,NOCLS,TOTSUR,TOTFLD,TOTVRT,NOCL,NVRT
*      ,NXTCLS,NOFEAT,MAXCLS,FETVEC(30),SYMMTX(62)
*      ,VARSIZ,STATKY,ISOKEY,MAPFMT,MAPKEY,SEQUEN(20),PERCEN,SIMERP
*      ,TORDEP,INUNIT,INFILE,INITM,PMIN,SURVEC(62),NOSUR2,CHNVC(30)
*      ,NOCHAN,FRCOMP,NOSEQ,MEAND0,MEANDU,
*      SYMD0,SYMDU,ITRIG0,ITRIGU,DOFLAG,
*      DUFLAG,DODU,STDOTS(60),NSDOTS,SUNCOR(30),LLNCAT,
*      OVERT(250,2),DRECT(60,2),DVPNT(11,2),IDCNT(2),NDOU(2)
*      ,MXFET1,MXP0P
REAL SUNCOR
COMMON/GLOBAL/HEAD(63),MAPTAP,DATAP,SAVTAP,BMFILE,BMKEY,
*      HISFIL,HISKEY,TRFORM,ERIPTP,ERPKEY,MAPUNT,NOFILE,
*      DRUMAD,DRMWDS,PAGSIZ,DATFIL,STAFIL,ASAV,ASAVFL
*      ,NHSTUN,NHSTFI,STRUN,MAPFIL
*      ,DOTUNT,DOTFIL,NCHPAS,TRNSFL,BMTRFL,HISTFL,PCHUNT,
*      CRDUNT,PRUNT,PANDIO
CSEND
DIMENSION IPLACE(NOPTS)
DIMENSION MEANS(NOFAT,MAXCLS),STDEV(NOFAT,MAXCLS)
DIMENSION SYMBLS(1),FL(12)
DIMENSION FLDINF(1)
DIMENSION CLD(MAXCLS,1),N(MAXCLS),NBLK(62),FINF(6)
REAL MEANS,STDEV,CLD
DIMENSION COL(3,110),OUT(110)
EQUIVALENCE (FINF(1),LINSTR), (FINF(4),SAMSTR),
*      (FINF(2),LINEND), (FINF(5),SAMEND),
*      (FINF(3),LININC), (FINF(6),SAMINC)
EQUIVALENCE (SYMMTX,SYMBLS)
EQUIVALENCE (COL(1,1),OUT(1)), (COL(1,111),NBLK)
DATA BLANK/' '/
IF (DODU.EQ.0) GO TO 20
SAVER=SYMMTX(LNCAT-DODU+DOFLAG)
SAVER=SYMMTX(LNCAT-DODU+DODU)
IF (DOFLAG.NE.0) SYMMTX(LNCAT-DODU+1) = SYMD0
IF (DUFLAG.NE.0) SYMMTX(LNCAT) = SYMDU
20  CONTINUE
WRITE (6,HEAD)
IF (KKT.GT.0) WRITE (6,240) KKT
IF (KKT.LT.0) WRITE (6,245) CLSNAM
WRITE (6,250) LNCAT
TOTPTS=TOTWRD/NOFEAT
WRITE (6,260) TOTPTS
WRITE (6,270)
DO 30 J=1,LNCAT
WRITE (6,280) J,SYMBLS(J),N(J)
30  CONTINUE
WRITE (6,290)
ISTART = 1
IEND = 12
LOOPCT = NOFEAT / 12
LOOPC1 = MOD(NOFAT,12)
IF (LOOPC1.GT.0) LOOPCT = LOOPCT + 1
IF (LOOPCT.EQ.1) IEND = NOFEAT
DO 45 M=1,LOOPCT
WRITE (6,300) (BLANK,FETVEC(J),J=ISTART,IEND)
DO 40 J=1,LNCAT
WRITE (6,310) J,(MEANS(I,J),I=ISTART,IEND)
40  CONTINUE
WRITE (6,315)
315 FORMAT(' ')
ISTART = IEND + 1
IEND = ISTART + IEND - 1
IF (IEND.GT. NOFEAT) IEND = NOFEAT
45  CONTINUE
ISTART = 1
IEND = 12
IF (LOOPCT.EQ.1) IFND = NOFEAT
WRITE (6,320)
DO 55 M=1,LOOPCT
WRITE (6,300) (BLANK,FETVEC(J),J=ISTART,IEND)
DO 50 J=1,LNCAT
```

PRI00010
PRI00020
PRI00030
PRI00040
PRI00050
PRI00060
PRI00070
PRI00080
PRI00090
PRI00100
PRI00110
PRI00120
PRI00130
PRI00140
PRI00150
PRI00160
PRI00170
PRI00180
PRI00190
PRI00200
PRI00210
PRI00220
PRI00230
PRI00240
PRI00250
PRI00260
PRI00270
PRI00280
PRI00290
PRI00300
PRI00310
PRI00320
PRI00330
PRI00340
PRI00350
PRI00360
PRI00370
PRI00380
PRI00390
PRI00400
PRI00410
PRI00420
PRI00430
PRI00440
PRI00450
PRI00460
PRI00470
PRI00480
PRI00490
PRI00500
PRI00510
PRI00520
PRI00530
PRI00540
PRI00550
PRI00560
PRI00570
PRI00580
PRI00590
PRI00600
PRI00610
PRI00620
PRI00630
PRI00640
PRI00650
PRI00660
PRI00670
PRI00680
PRI00690
PRI00700
PRI00710
PRI00720
PRI00730
PRI00740
PRI00750
PRI00760
PRI00770
PRI00780
PRI00790

FILE: PRINT

```

WRITE (6,310) J, (STDEV(I,J), I=ISTART, IEND)
50 CONTINUE
WRITE (6,315)
  ISTART = IEND + 1
  IEND = ISTART + IEND - 1
  IF (IEND .GT. NOFEAT) IEND = NOFEAT
55 CONTINUE
  L=1
  J=LNCAT
  IF (J.GT.15) J=15
  60 WRITE (6,340) (K,K=L,J)
  DO 70 I=1,LNCAT
  70 WRITE (6,350) I, (CLD(I,K), K=L,J)
    IF (J.EQ.LNCAT) GO TO 80
    L=L+15
    J=J+15
    IF (J.GE.LNCAT) J=LNCAT
    GO TO 60
  80 CONTINUE
    IF (KKT.EQ.-1) GO TO 90
    IF (MOD(KKT, MAP).NE.0) RETURN
  90 CONTINUE
    IRC=IR0
    ICCT=NOPTS
    IF (IRD.EQ.0) ICCT=KPTS
    IF (IRD.EQ.0) GO TO 110
    ADRES2=HEGIN2
    CALL RREAD(ADRES2, IPLACE, ICCT, ISTAT)
  105 IF (ISTAT.EQ.1) GO TO 105
    ADRES2=ADRES2+ICCT
    IRFC=1
  110 JPTS=0
    IRC=IPC-1
    CALL SFTMFG(66,0,66)
    IV=5
    DO 200 IFLD=1,NOFLD
  C* 200 IFLD=1,NOFLD
    200 IFLD=1,NOFLD
    DO 115 I=1,LNCAT
  115 NRLK(I)=0
    FLNAM = FLDINF(IV)
    NV= FLDINF(IV+1)
    IR=IV+2*NV+2
    DO 116 I=1,6
  116 FINE(I)=FLDINF(IR+I)
    J=0
  120 DO 130 I=SAMSTR,SAMEND,SAMINC
    J=J+1
    COL(1,J)=1/100
    COL(2,J)=MOD(I,100)/10
    COL(3,J)=MOD(I,10)
    IF (J.EQ.110) GO TO 140
  130 CONTINUE
  140 LPTS=J
    WRITE (6,220)
    WRITE (6,HEAD)
    TPTS=FLDINF(IV+NV*2+2)
    WRITE (6,330) FLNAM,TPTS
    DO 150 I=1,3
  150 WRITE (6,210) (COL(I,J), J=1,LPTS)
    WRITE (6,220)
    DO 180 I=LINSTR,LINEND,LININC
  C* FIND FIELD INTERSECTIONS FOR THIS LINE
    CALL FOLINT(FLDINF(IV+2), NV, FL, I, PTS, NFL)
    DO 155 J=1,110
  155 OUT(J)=HLANK
    DO 175 IJ=1,NFL,2
    IR=(FL(IJ)-SAMSTR)/SAMINC + 1
    IF=(FL(IJ+1)-SAMSTR)/SAMINC + 1
    IF (MOD(SAMSTR,SAMINC).NE.MOD(FL(IJ),SAMINC)) IB=IB+1
    IF (IR.GT.IE) GO TO 175
    DO 170 J=IR,IE
    JPTS=JPTS+1
    IF (JPTS.LE.ICCT) GO TO 160
    IF (IRC.EQ.1) ICCT=KPTS
    CALL RREAD(ADRES2, IPLACE, ICCT, ISTAT)
    ADRES2=ADRES2+ICCT
  156 IF (ISTAT.EQ.1) GO TO 156
    IRC=IPC-1
    JPTS=1

```

PRI00800
 PRI00810
 PRI00820
 PRI00830
 PRI00840
 PRI00850
 PRI00860
 PRI00870
 PRI00880
 PRI00890
 PRI00900
 PRI00910
 PRI00920
 PRI00930
 PRI00940
 PRI00950
 PRI00960
 PRI00970
 PRI00980
 PRI00990
 PRI01000
 PRI01010
 PRI01020
 PRI01030
 PRI01040
 PRI01050
 PRI01060
 PRI01070
 PRI01080
 PRI01090
 PRI01100
 PRI01110
 PRI01120
 PRI01130
 PRI01140
 PRI01150
 PRI01160
 PRI01170
 PRI01180
 PRI01190
 PRI01200
 PRI01210
 PRI01220
 PRI01230
 PRI01240
 PRI01250
 PRI01260
 PRI01270
 PRI01280
 PRI01290
 PRI01300
 PRI01310
 PRI01320
 PRI01330
 PRI01340
 PRI01350
 PRI01360
 PRI01370
 PRI01380
 PRI01390
 PRI01400
 PRI01410
 PRI01420
 PRI01430
 PRI01440
 PRI01450
 PRI01460
 PRI01470
 PRI01480
 PRI01490
 PRI01500
 PRI01510
 PRI01520
 PRI01530
 PRI01540
 PRI01550
 PRI01560
 PRI01570
 PRI01580

FILE: PRINT

```

160 CONTINUE
   K=IPLACE(JPTS)
   NBLK(K)=NBLK(K)+1
   LINE=I
   IF (J.GT.110) GO TO 170
   OUT(J)=SYMBLS(K)
170 CONTINUE
175 CONTINUE
   WRITE (6,230)LINE,(OUT(J),J=1,LPTS)
180 CONTINUE
   IV=IV + NV*2 + 9
   WRITE (6,370)
   DO 190 I=1,LNCAT
190  WRITE (6,380)I,SYMBLS(I),NBLK(I)
200 CONTINUE
   IF (DODU.EQ.0) GO TO 205
   SYMMTX(LNCAT-DODU+DOFLAG)=SAVEP
   IF (DOFLAG.NE.0) SYMMTX(LNCAT) = SAVEB
205 CONTINUE
   CALL SETMPG(66,4,62)
   RETURN
210 FORMAT(9X,11011)
220 FORMAT(/)
230 FORMAT(2X,15,2X,110A1)
240 FORMAT(///' INTERMEDIATE PRINTOUT FOR ITERATION',I5//)
245 FORMAT(///' FINAL CLUSTER SUMMARY FOR CLASS',1X,A4//)
250 FORMAT(///' TOTAL NUMBER OF CLUSTERS =',I3)
260 FORMAT(///' TOTAL NUMBER OF POINTS =',I7)
270 FORMAT(///' CLUSTER      SYMBOL      POINTS IN CLUSTER')
280 FORMAT(4X,12,9X,A1,10X,I7)
290 FORMAT(///15X,'MEANS'//)
300 FORMAT(//2X,'CLUSTER',5X,12(A1,'CH(',12,')',1X))
310 FORMAT(5X,12,7X,12(F7.2,1X))
320 FORMAT(///10X,' STANDARD DEVIATIONS'//)
330 FORMAT(///2X,A4,///' TOTAL NUMBER OF POINTS IN THIS FIELD',I7/
*)
340 FORMAT(///15X,'DISTANCES BETWEEN CLUSTERS'//1X,'CLUSTER',15I8)
350 FORMAT(3X,12,5X,15F8,2)
370 FORMAT(///2X,'POINTS PER CLUSTER IN THIS FIELD'/3X,'CLUSTER',
* 5X,'SYMBOL',5X,'POINTS'//)
380 FORMAT(6X,12,10X,A1,7X,I5)
END

```

```

PRI01590
PRI01600
PRI01610
PRI01620
PRI01630
PRI01640
PRI01650
PRI01660
PRI01670
PRI01680
PRI01690
PRI01700
PRI01710
PRI01720
PRI01730
PRI01740
PRI01750
PRI01760
PRI01770
PRI01780
PRI01790
PRI01800
PRI01810
PRI01820
PRI01830
PRI01840
PRI01850
PRI01860
PRI01870
PRI01880
PRI01890
PRI01900
PRI01910
PRI01920
PRI01930
PRI01940
PRI01950
PRI01960
PRI01970
PRI01980
PRI01990
PRI02000

```

FILE: PRTCOV

| | |
|---|--|
| <pre> SUBROUTINE PRTCOV(COVMTX,AVEMTX,CV1,AV1,CLSMTX) C WRITE HEADING FOR TRANSFORMED COVARIANCE MATRIX C IMPLICIT INTEGER(A-Z) REAL COVMTX(CV1,NOSUB2),AVEMTX(AV1,NOSUB2) C INCLUDE COMRK1.LIST INCLUDE COMRK4.LIST INCLUDE COMRK6.LIST INCLUDE COMRK9.LIST COMMON/INFORM/NOCLS2,NOSUB2,NOFET2,VARSZ2,TOTVT2,NOFLD2, * AVAP2,COVAR2,CLS1D2,SURN02,SURDS2,FLOSV2,VERTX2, * FETVC2(30),SURVC2(75),SURPTR(75),CLSV2(60), * KFPPTS(40),NOGRP,GRPNAM(60),GRPDEX(61), * GRPCHK(61),GROUPS(124) DIMENSION HED1(15),HED2(15),DATE(3),COMENT(15) EQUIVALENCE (HED1(1),HEAD(4)),(DATE(1),HEAD(22)), 2 (HED2(1),HEAD(30)),(COMENT(1),HEAD(48)) COMMON/GLOBAL/HEAD(63),MAPTAP,DATAP,SAVTAP,RMFILE,RMKEY, * HISFIL,HISKEY,TRFORM,ERPTP,ERPKEY,MAPUNT,NOFILE, * DRUMAD,DRUMDS,PAGSIZ,DATFIL,STAFIL,ASAV,ASAVL * .NHSTUN,NHSTFI,SCTRUN,MAPFIL * .DOTUNT,DOTFIL,NCHPAS,TRNSFL,BMTRFL,HISTFL,PCHUNT, * CHDUNT,PRTUNT,RANDIO C DATA TRANSFORMATION COMMON BLOCK COMMON/TRBLCK/OUTFMT,NOFEAT,FLDINF(6), FETVEC(30) CSEND C DIMENSION CLSMTX(NOSUB2) C DATA BCDTWO/'2'/ C CNT=7+(5+3+2*AV1)*((AV1+11)/12) CNT=PAGSIZ/CNT INC=CNT DO 1 ICLAS=1,NOSUB2 IF (INC.LT.CNT) GO TO 2 WRITE(6,HEAD) INC=0 2 WRITE(6,3) CLSMTX(ICLAS) 3 FORMAT(/' SUBCLASS ',A4) DO 4 LOC=1,AV1,12 STOP=LOC+11 IF (STOP.GT.AV1) STOP=AV1 4 WRITE(6,5) (AVEMTX(I,ICLAS),I=LOC,STOP) 5 FORMAT(' MEAN',3X,12F9.2) WRITE(6,6) 6 FORMAT(' COVARIANCE MATRIX') CALL WRTMTX(COVMTX(1,ICLAS),AV1,BCDTWO) INC=INC+1 1 CONTINUE RETURN END </pre> | <pre> PRT00010 PRT00020 PRT00030 PRT00040 PRT00050 PRT00060 PRT00070 PRT00080 PRT00090 PRT00100 PRT00110 COM00010 COM00020 COM00030 COM00040 COM00050 COM00060 COM00070 COM00080 COM00090 COM00100 COM00110 COM00120 COM00130 COM00140 COM00150 COM00160 COM00170 COM00180 COM00190 COM00200 COM00210 COM00220 COM00230 COM00240 COM00250 COM00260 COM00270 COM00280 COM00290 COM00300 COM00310 COM00320 COM00330 COM00340 COM00350 COM00360 COM00370 COM00380 COM00390 </pre> |
|---|--|

FILE: RANK

```

SUBROUTINE RANK(NOFAT,FETVC2,LNCAT,MEANS,IPTT)
  IMPLICIT INTEGER(A-X)
  * REAL MEANS(NOFAT,LNCAT).SAVE,G(60)
  DIMENSION FETVC2(26),IPTT(LNCAT)
  C INCLUDE COMRA4
  COMMON/GLOBAL/HEAD(63),MAPTAP,DATAP,SAVTAP,RMFILE,BMKEY,
  * HISFIL,HISKEY,TRFORM,ERIPTP,EPPKEY,MAPUNT,NOFILE,
  * DRUMAN,DRMWDOS,PAGSIZ,DATFIL,STAFIL,ASAV,ASAVFL
  * ,NHSTIN,NHSTFI,SCTRUN,MAPFIL
  * ,DOTUNT,DOTFIL,NCHPAS,TRNSFL,BMTRFL,HISTFL,PCHUNT,
  * CRDUNT,PRUNT,RANDIO
CSEND
  IF (MOD(NOFAT,4))10,20,10
10  WRITE(6,500)
    GO TO 99
20  DO 30 I=1,LNCAT
    IPTT(I)=1
30  CONTINUE
    DO 40 J=1,LNCAT
    G(J)=0
40  CONTINUE
    DO 50 I=1,NOFAT,NCHPAS
    DO 50 J=1,LNCAT
    G(J)=G(J)+(-0.24*MEANS(I,J)-0.56*MEANS(I+1,J)+0.6*MEANS(I+2,J)
    *0.49*MEANS(I+3,J))
50  CONTINUE
    J=0
60  J=J+1
    IF (J.GT.LNCAT)GO TO 90
    IF (J.EQ.LNCAT)GO TO 75
    IF (G(J).LT.G(J+1))GO TO 70
    GO TO 60
70  SAVF=G(J)
    G(J)=G(J+1)
    G(J+1)=SAVF
    ISAVE=IPTT(J)
    IPTT(J)=IPTT(J+1)
    IPTT(J+1)=ISAVE
75  K=J
80  IF (K.EQ.1)GO TO 60
    IF (G(K).LT.G(K-1))GO TO 60
    SAVF=G(K-1)
    G(K-1)=G(K)
    G(K)=SAVF
    ISAVE=IPTT(K-1)
    IPTT(K-1)=IPTT(K)
    IPTT(K)=ISAVE
    K=K-1
    GO TO 80
90  CONTINUE
    WRITE(6,510) (I,IPTT(I),G(I),I=1,LNCAT)
99  CONTINUE
    RETURN
500  FORMAT(1X,'THE NUMBER OF CHANNELS ARE NOT A MULTIPLE OF 4.
  *THE COLOR KEYS WILL BE ORDERED BY CLUSTER NUMBER.')
510  FORMAT(1X,'COLOR KEY ',12,' IS CLUSTER ',12,' GREENNESS=',F7.2/)
  END

```

RAN00010
 RAN00020
 RAN00030
 RAN00040
 RAN00050
 COM00010
 COM00020
 COM00030
 COM00040
 COM00050
 COM00060
 RAN00070
 RAN00080
 RAN00090
 RAN00100
 RAN00110
 RAN00120
 RAN00130
 RAN00140
 RAN00150
 RAN00160
 RAN00170
 RAN00180
 RAN00190
 RAN00200
 RAN00210
 RAN00220
 RAN00230
 RAN00240
 RAN00250
 RAN00260
 RAN00270
 RAN00280
 RAN00290
 RAN00300
 RAN00310
 RAN00320
 RAN00330
 RAN00340
 RAN00350
 RAN00360
 RAN00370
 RAN00380
 RAN00390
 RAN00400
 RAN00410
 RAN00420
 RAN00430
 RAN00440
 RAN00450
 RAN00460
 RAN00470
 RAN00480
 RAN00490
 RAN00500
 RAN00510
 RAN00520

FILE: RDDOTS

```

C      UTILITY ROUTINE THAT READS THE DOTFIL
C
C      SUBROUTINE RDDOTS(DOTS,DOTVEC,TOTDT3,TYPSWT,SIZES,TOTDT2,
C      * NOCAT,CATNAM,NOFEAT2,FETVC2,NOFEAT,FETVEC,NOSUN,ANGLE,
C      * NOFLD,TOTVRT,FLDSAV,VERTEX,/KVAR/)
C      IMPLICIT INTEGER(A-Z)
C      DIMENSION KVAR(SIZES,1)
C      REAL KVAR
C
C      TYPSWT = 1 -RETURNS SPECTRAL INFO
C      TYPSWT = 2 -RETURNS SPATIAL INFO
C      TYPSWT = 3 -RETURNS BOTH SPATIAL AND SPECTRAL INFO
C
C      DATA BLANK/' '/
C      INCLUDE COMHKG.LIST
C      COMMON/GLOBAL/HEAD(63),MAPTAP,DATAPE,SAVTAP,RMFILE,RMKEY,
C      * HISFIL,HISKEY,TRFORM,ERIPTP,ERPKEY,MAPUNT,NOFILE,
C      * DRUMAD,DRMADS,PAGSIZ,DATEIL,STAFIL,ASAV,ASAVFL,
C      * NHSTJN,NHSTFI,SCTRUN,MAPFIL,
C      * DOTUNT,DOTFIL,NCHPAS,TRNSFL,BMTRFL,HISTFL,PCHUNT,
C      * CRDUNT,PRUNT,HANDIO
C
C      DIMENSION CATNAM(1),DOTVEC(1),DOTS(SIZES,1)
C      DIMENSION FETVEC(30),FETVC3(70),FETVC2(1)
C      DIMENSION FLDSAV(4,1),VERTEX(2,1),ANGLE(1)
C      DIMENSION TEMDOT(5000)
C
C      READ REC NO. 1 FOR INDICES
C
C      REWIND DOTUNT
C      CALL FSRFSL(DOTUNT,DOTFIL,ISTAT)
C
C      READ(DOTUNT)NOCAT,NOFEAT,NOFLD,TOTVRT,TOTDOT,NOSUN,(CATNAM(I),
C      * I=1,NOCAT),SIZE
C
C      COMPUTE ADDRESSES FOR ARRAY
C
C      DOTS1 = 1
C
C      READ REC. NO. 2
C
C      IF (TYPSWT.EQ. 1) READ(DOTUNT)(FETVEC(I),I=1,NOFEAT)
C      IF (TYPSWT.EQ. 2) READ(DOTUNT) DUMMY
C      IF (TYPSWT.EQ. 3) READ(DOTUNT)(FETVEC(I),I=1,NOFEAT)
C      * ((FLDSAV(I,J),I=1,4),J=1,NOFLD),((VERTEX(I,J),I=1,2),
C      * J=1,TOTVRT),(ANGLE(I),I=1,NOSUN)
C
C      IF (TYPSWT.EQ. 2) GO TO A7
C      IF (NOFEAT2.NE. 0) GO TO 60
C
C      SET DEFAULT CHANNELS
C
C      DO 50 I=1,NOFEAT
C      FETVC2(I) = 1
C      FETVC3(I) = 1
C      NOFEAT2 = NOFEAT
C      GO TO A7
C      DO 60 J=1,NOFEAT2
C      DO 70 K=1,NOFEAT
C      IF (FETVEC(K).EQ. FETVC2(J)) GO TO 75
C      70 CONTINUE
C      WRITE(*,85)FETVC2(J),(FETVEC(I),I=1,NOFEAT)
C      85 FORMAT('/// CHANNEL ',I2,' IS NOT ON DOTFIL'//) CHANNELS ARE',30I3)
C      CALL CHH44
C      75 FETVC3(J) = K
C
C      90 CONTINUE
C      A7 IF (TOTDT3.EQ. 0) GO TO 96
C
C      *** CODE ADDED NOV 21, 1978 FOR LIST PROCESSING
C      IF (TOTDT3.GT.TOTDOT) TOTDT3 = TOTDOT
C
C      DO 95 J=1,TOTDT3

```

RDD00010
 RDD00020
 RDD00030
 RDD00040
 RDD00050
 RDD00060
 RDD00070
 RDD00080
 RDD00090
 RDD00100
 RDD00110
 RDD00120
 RDD00130
 RDD00140
 RDD00150
 RDD00160
 RDD00170
 RDD00180
 RDD00190
 RDD00200
 RDD00210
 RDD00220
 RDD00230
 RDD00240
 RDD00250
 RDD00260
 RDD00270
 RDD00280
 RDD00290
 RDD00300
 RDD00310
 RDD00320
 RDD00330
 RDD00340
 RDD00350
 RDD00360
 RDD00370
 RDD00380
 RDD00390
 RDD00400
 RDD00410
 RDD00420
 RDD00430
 RDD00440
 RDD00450
 RDD00460
 RDD00470
 RDD00480
 RDD00490
 RDD00500
 RDD00510
 RDD00520
 RDD00530
 RDD00540
 RDD00550
 RDD00560
 RDD00570
 RDD00580
 RDD00590
 RDD00600
 RDD00610
 RDD00620
 RDD00630
 RDD00640
 RDD00650
 RDD00660
 RDD00670
 RDD00680
 RDD00690
 RDD00700
 RDD00710
 RDD00720
 RDD00730
 RDD00740
 RDD00750
 RDD00760
 RDD00770
 RDD00780
 RDD00790

FILE: RDDOTS

| | | |
|---|---|-----------|
| | IF (DOTVEC(J).LE.TOTDOT) GO TO 95 | RDD000800 |
| | WRITE(4,90)DOTVEC(J).TOTDOT | RDD000810 |
| | 90 FORMAT(/ 'DOT NO. '13.' IS NOT ON DOTFIL' / ' FILE CONTAINS '13, | RDD000820 |
| | 'DOTS') | RDD000830 |
| | CALL CMERR | RDD000840 |
| | 95 CONTINUE | RDD000850 |
| | 96 CONTINUE | RDD000860 |
| C | | RDD000870 |
| C | READ REC NO.3 -- (DOTS | RDD000880 |
| C | TOTAL = SIZE * TOTDOT | RDD000890 |
| C | READ(DOTUNT)(TEMDOT(DOTS1-1+I),I=1,TOTAL) | RDD000900 |
| C | | RDD000910 |
| C | IF (TYPST .EQ. 1) GO TO 130 | RDD000920 |
| C | IF (TYPST .EQ. 2) GO TO 150 | RDD000930 |
| C | IF (TYPST .EQ. 3) GO TO 180 | RDD000940 |
| C | | RDD000950 |
| C | RETRIEVE SPECTRAL INFO | RDD000960 |
| C | | RDD000970 |
| C | 130 CONTINUE | RDD000980 |
| C | SIZES = NOFET2 | RDD000990 |
| C | TOTDT2 = TOTDT3 | RDD001000 |
| C | CALL RDDDT1(TEMDOT,DOTS,KVAR,SIZES,TOTDT2,DOTVEC,FETVC3, | RDD001010 |
| C | 'SIZE,TOTDOT,TOTDT3,NOFET2,TYPST) | RDD001020 |
| C | RETURN | RDD001030 |
| C | | RDD001040 |
| C | RETRIEVE SPATIAL INFO | RDD001050 |
| C | | RDD001060 |
| C | 150 CONTINUE | RDD001070 |
| C | SIZES = 4 | RDD001080 |
| C | TOTDT2 = TOTDOT | RDD001090 |
| C | | RDD001100 |
| C | CALL RDDDT1(TEMDOT,DOTS,KVAR,SIZES,TOTDT2,DOTVEC,FETVC3, | RDD001110 |
| C | 'SIZE,TOTDOT,TOTDT3,NOFET2,TYPST) | RDD001120 |
| C | RETURN | RDD001130 |
| C | | RDD001140 |
| C | 180 CONTINUE | RDD001150 |
| C | | RDD001160 |
| C | RETRIEVE SPECTRAL AND SPATIAL INFO | RDD001170 |
| C | | RDD001180 |
| C | SIZES = SIZE | RDD001190 |
| C | TOTDT2 = TOTDOT - TOTDT3 | RDD001200 |
| C | CALL RDDDT1(TEMDOT,DOTS,KVAR,SIZES,TOTDT2,DOTVEC,FETVC3, | RDD001210 |
| C | 'SIZE,TOTDOT,TOTDT3,NOFET2,TYPST) | RDD001220 |
| C | RETURN | RDD001230 |
| C | | RDD001240 |
| C | END | RDD001250 |
| | | RDD001260 |
| | | RDD001270 |

FILE: RDN0T1

| | | |
|---|---|----------|
| | SUBROUTINE RDN0T1(TEMDOT,DOTS,KVAR,SIZES,TOTDT2,DOTVEC,FETVC3,
SIZE,TOTDOT,TOTDT3,NOFET2,TYPSWT) | R0000010 |
| C | IMPLICIT INTEGER (A-Z) | R0000020 |
| | DIMENSION TEMDOT(1),DOTS(SIZES,1),DOTVEC(1),FETVC3(1) | R0000030 |
| | REAL KVAR(SIZES,1) | R0000040 |
| C | TYPSWT = 1 -- RETRIEVE SPECTRAL INFO | R0000050 |
| C | = 2 -- RETRIEVE SPATIAL INFO | R0000060 |
| C | = 3 -- RETRIEVE SPECTRAL AND SPATIAL INFO | R0000070 |
| C | GO TO (130,150,180),TYPSWT | R0000080 |
| C | RETRIEVE SPECTRAL INFO | R0000090 |
| C | 130 CONTINUE | R0000100 |
| C | PICK SURSET OF DOTS AND CHANNELS | R0000110 |
| C | DO 140 K=1,TOTDT2 | R0000120 |
| | KK = DOTVEC(K) | R0000130 |
| | DO 140 J=1,SIZES | R0000140 |
| | JJ = FETVC3(J) | R0000150 |
| | JJJ = (KK-1)*SIZE + JJ+4 | R0000160 |
| | KVAR(J,K) = FLOAT(TEMDOT(JJJ)) | R0000170 |
| C | 140 CONTINUE | R0000180 |
| | RETURN | R0000190 |
| C | RETRIEVE SPATIAL INFO | R0000200 |
| C | 150 CONTINUE | R0000210 |
| | DO 170 I=1,TOTDT2 | R0000220 |
| | KK = (I-1)*SIZE | R0000230 |
| | DO 170 K=1,SIZES | R0000240 |
| | 170 DOTS(K,I) = TEMDOT(KK+K) | R0000250 |
| C | RETURN | R0000260 |
| C | 180 CONTINUE | R0000270 |
| C | RETRIEVE SPECTRAL AND SPATIAL INFO | R0000280 |
| C | JJ = 0 | R0000290 |
| | KK = 1 | R0000300 |
| | DO 300 K=1,TOTDOT | R0000310 |
| | IF (KK.GT. TOTDT3) GO TO 190 | R0000320 |
| | IF (K.NE. DOTVEC(KK)) GO TO 190 | R0000330 |
| | KK = KK+1 | R0000340 |
| | GO TO 300 | R0000350 |
| | 190 JJ = JJ + 1 | R0000360 |
| | JJJ = (K-1)*SIZE | R0000370 |
| | DO 200 J=1,4 | R0000380 |
| | 200 DOTS(J,JJ) = TEMDOT(JJJ+J) | R0000390 |
| C | DO 210 J=1,NOFET2 | R0000400 |
| | I = FETVC3(J) | R0000410 |
| | 210 DOTS(4+J,JJ) = TEMDOT(JJJ+4+I) | R0000420 |
| | 300 CONTINUE | R0000430 |
| C | RETURN | R0000440 |
| C | END | R0000450 |
| | | R0000460 |
| | | R0000470 |
| | | R0000480 |
| | | R0000490 |
| | | R0000500 |
| | | R0000510 |
| | | R0000520 |
| | | R0000530 |
| | | R0000540 |
| | | R0000550 |
| | | R0000560 |
| | | R0000570 |
| | | R0000580 |
| | | R0000590 |
| | | R0000600 |
| | | R0000610 |

ۛۛۛۛۛ

THIS SUBROUTINE READS THE 'MEAN' CARD DECK OR FILE FOR ISOCLS.

```

DIMENSION HED1(15),HED2(15),DATE(3),COMENT(15)
EQUIVALENCE (HED1(1),HEAD(4)),(DATE(1),HEAD(22)),
              (HED2(1),HEAD(30)),(COMENT(1),HEAD(48))

```

```
COMMON/PASS/STOP, LNCAT, NMIN, KRN, STDMAX, DLMIN, SEP,  
MAP, SPTRIG, IRD, KPTS, NOPTS, PUNCH,  
ICHN, CHNTHS, ICHAIN(62), NWDS, IREGIN, REGIN),  
RFGIN2, RFGIN3, CLSNAM, NOFLD, IPT, TOTWRD, TOTPTS,  
NCLASS, NOCLS, TOTSUB, TOTFLD, TOTVRT, NOCL, NVRT  
NXTCLS, NOFFAT, MAXCLS, FETVEC(30), SYMMTX(62)  
VARSIZ, STATKY, ISOKEY, MAPFMT, MAPKEY, SEQUEN(20), PERCEN, SIMERP  
TOPDER, INUNIT, INFILE, INITM, PMIN, SURVEC(62), NOSUB2, CHNVC(30)  
NOCHAN, ERCOMP, NOSEQ, MEAND0, MEANDU,  
SYMNO, SYMDU, ITRIG0, ITRIGU, DOFLAG,  
DUFLAG, DODU, SDOTS(60), NSDOTS, SUNCOR(30), LLNCAT,  
DVRT(250,2), DRECT(60,2), DVPNT(11,2), IDCNT(2), NDOU(2)  
MXFFT1, MAXPOP  
REAL SUNCOR  
COMMON/GLOBAL/HEAD(63), MAPTAP, DATAPE, SAVTAP, BMFILE, BMKEY,  
HISFIL, HISKEY, TRFORM, ERPTP, ERPKEY, MAPUNT, NOFILE,  
DRUMAU, DRUMDS, PAGESIZ, DATFIL, STAFIL, ASAV, ASAVFL  
NHSTUN, NHSTFI, SCTRUN, MAPFIL  
DOTUNT, DOTFIL, NCHPAS, TRNSFL, BMTRFL, HISTFL, PCHUNT,  
CRDUNT, PRUNT, RANDIO
```

CSEND

```
COMMON/PASSA/NOFET1,FTVEC1(30)
DIMENSION MENS(30,1)
READ(CRDUNT,500) LNCAT,NOFET1,(FTVEC1(I), I=1,NOFET1)
DO 10 I=1,LNCAT
  READ(CRDUNT,510) (MENS(J,I), J=1,NOFET1)
  FORMAT(5X,15,15X,15/5X,30I2)
  FORMAT(5X,5F15,8)
  ADDRES=1BEGIN
  CALL PWRITE(ADDRES,LNCAT,1,LSTAT)
  ADDRES=ADDRES+1
  CALL RWRITE(ADDRES,NOFET1,1,LSTAT)
  ADDRES=ADDRES+1
  CALL RWRITE(ADDRES,FTVEC1,NOFET1,LSTAT)
  ADDRES=ADDRES+NOFET1
  KW=MXFFT1*LNCAT
  CALL RWRITE(ADDRES,MENS,KW,JSTAT)
  IF(JSTAT.EQ.1)GOTO 11
RETURN
```

22

```
ENTRY RFILE(MFANS,MENS)
DIMENSION MEANS(NOFAT,MAXCLS)
ADDRESS=IBEGIN
CALL RREAD(ADDRESS,LNCAT,1,LSTAT)
ADDRESS=ADDRESS+1
CALL RREAD(ADDRESS,NOFAT1,1,LSTAT)
ADDRESS=ADDRESS+1
CALL RREAD(ADDRESS,FIVEC1,NOFAT1,LSTAT)
ADDRESS=ADDRESS+NOFAT1
KW=MAXFAT1*LNCAT
CALL RREAD(ADDRESS,MENS,KW,JSTAT)
IF(JSTAT.EQ.1)GOTO 12
```

STORE ONLY CHANNELS REQUESTED IN FETVEC

```

      IF(LNCAT.GT.MAXCLS)LNCAT=MAXCLS
      DO 40 J=1,NOFFAT
      DO 30 K=1,NOFFAT
      IF(FEVEEC(J).NE.FEVEEC(K))GO TO 30
      DO 20 I=1,LNCAT
20    MEANS(J,I) = MEANS(K,I)
      GO TO 40
30    CONTINUE
      WRITE(6,100)FEVEEC(J)
      DO 35 I=1,LNCAT
35    MEANS(J,I) = 50. + I*10.
40    CONTINUE

```

RDM00010
 RDM00020
 RDM00030
 RDM00040
 RDM00050
 RDM00060
 RDM00070
 RDM00080
 COM00010
 COM00020
 COM00030
 COM00010
 COM00020
 COM00030
 COM00040
 COM00050
 COM00060
 COM00070
 COM00080
 COM00090
 COM00100
 COM00110
 COM00120
 COM00130
 COM00140
 COM00010
 COM00020
 COM00030
 COM00040
 COM00050
 COM00060
 RDM00100
 RDM00110
 RDM00120
 RDM00130
 RDM00140
 RDM00150
 RDM00160
 RDM00170
 RDM00180
 RDM00190
 RDM00200
 RDM00210
 RDM00220
 RDM00230
 RDM00240
 RDM00250
 RDM00260
 RDM00270
 RDM00280
 RDM00290
 RDM00300
 RDM00310
 RDM00320
 RDM00330
 RDM00340
 RDM00350
 RDM00360
 RDM00370
 RDM00380
 RDM00390
 RDM00400
 RDM00410
 RDM00420
 RDM00430
 RDM00440
 RDM00450
 RDM00460
 RDM00470
 RDM00480
 RDM00490
 RDM00500
 RDM00510
 RDM00520
 RDM00530
 RDM00540
 RDM00550
 RDM00560
 RDM00570

FILE: RDMEAN

| | | |
|-----|---|----------|
| C* | | RDM00580 |
| C* | PRINT INITIAL CLUSTER CENTERS | RDM00590 |
| C* | | RDM00600 |
| | WRITE(6,200) | RDM00610 |
| | IR=1 | RDM00620 |
| | IF=12 | RDM00630 |
| 45 | IF(NOFFAT.LT.IE) IE=NOFEAT | RDM00640 |
| | WRITE(6,300) (FFIVEC(J),J=IB,IE) | RDM00650 |
| | DO 50 I=1,LNCAT | RDM00660 |
| 50 | WRITE(6,400) I, (MEANS(J,I),J=IB,IE) | RDM00670 |
| | IF(IE.EQ.NOFEAT) GO TO 60 | RDM00680 |
| | IR=IR+12 | RDM00690 |
| | IF=IF+12 | RDM00700 |
| | GO TO 45 | RDM00710 |
| 60 | RETURN | RDM00720 |
| 100 | FORMAT(' MEANS FOR CHANNEL',I4,' ARE NOT ON FILE--DUMMY VALVES WILL BE USED') | RDM00730 |
| 200 | FORMAT('///15X,' INITIAL CLUSTER MEANS'/) | RDM00740 |
| 300 | FORMAT(2X,'CLUSTER',2X,12(1X,'CH(',I2,')',1X)) | RDM00750 |
| 400 | FORMAT(5X,I2,5X,12(F6.2,2X)) | RDM00760 |
| | END | RDM00770 |
| | | RDM00780 |

19-70

443

CCOOPPY

AND WRITE SAVTAP FILE

CSFND

C

C

C

cc

C

3

1

C

C

C

2

C

~~19-71~~
444

FILE: RDMODK

```

      SUBD2 = FLDSV1 - 1
      WRITE(SAVTAP)(ARRAY(I),I=CLSID1,SUBD2)
C 7 CONTINUE
      MEAN2 = CLSID1 - 1
      DO 10 I=1,NOSUR
      READ(CRDUNT,240) KEPPTS(I)
      READ(CRDUNT,280) (AVAR(J),J=1,NOFEAT)
      READ(CRDUNT,280) (COVAR(J),J=1,VARSIZ)
C
      IF (ISTAT.GT. 0) GO TO 10
CI WRITE(SAVTAP)KEPPTS(I),(ARRAY(J),J=1,MEAN2)
  10 CONTINUE
CI IF (ISTAT.GT. 0) STAFIL = -1
   IF (ISTAT.GT. 0) RETURN
CI
      END FILE SAVTAP
      RETURN
C
  215 FORMAT(A4.6X,I2.8X,I2.8X,I2)
  220 FORMAT(10X,14I5)
  230 FORMAT(16X,9(2X,A4.2X)))
  240 FORMAT(17X,24(1X,I2)))
  250 FORMAT(18X,10(A4.3X)))
  260 FORMAT(12X,1A)
  270 FORMAT(6X,I2F6.2)
  280 FORMAT(5X,5E15.8)
      END
```

RDM00800
RDM00810
RDM00820
RDM00830
RDM00840
RDM00850
RDM00860
RDM00870
RDM00880
RDM00890
RDM00900
RDM00910
RDM00920
RDM00930
RDM00940
RDM00950
RDM00960
CIRDM00970
RDM00980
RDM00990
RDM01000
RDM01010
RDM01020
RDM01030
RDM01040
RDM01050
RDM01060
RDM01070
RDM01080
RDM01090

FILE: RENDAT

```

SUBROUTINE RENDAT(COVAR,AVAR,CLSDFS,SURNO,SURDES,FLDSAV,VERTEX,
*          COV,AVEN,CLSDFS,SURNOS,SURDS,FLDSV,VERTX,
*          NOFEAT,VARSIZ,NOCLS,NOFLD,NOSUB,FETVEC)
IMPLICIT INTEGER (A-Z)
DIMENSION FETVEC(30)

* * * * *

READS COVARIANCES AND MEANS FROM FILE AND REDUCES STATS
* * * * *

INCLUDE COMK1.LIST
INCLUDE COMK6.LIST
COMMON/INFOPH/NOCLS2,NOSUB2,NOFET2,VARSZ2,TOTVT2,NOFLD2,
*          AVAR2,COVAR2,CLSDF2,SURNO2,SURDS2,FLDSV2,VERTX2,
*          FETVC2(30),SURVC2(75),SUPPTR(75),CLSVC2(60),
*          KFPPTS(60),NOGRP,GRPNAM(60),GRPDEX(61),
*          GRFCHK(61),GROUPS(124)
COMMON/GLOBAL/HEAD(63),MAPTAP,DATAP,SAVTAP,BMFILE,BMKEY,
*          HISFIL,HISKEY,TRFORM,ERIPTR,ERPKEY,MAPUNT,NOFILE,
*          DRUMAD,DRMWDOS,PAGSIZ,DATFIL,STAFIL,ASAV,ASAVFL
*          ,NHSTUN,NHSTFI,STRUN,MAPFIL
*          ,DOTUNT,DOTFIL,NCHPAS,TRNSFL,BMTRFL,HISTFL,PCHUNT,
*          CRDUNT,PRUNT,RANDIO
CSEND
COMMON/RESTKN/KFPPTS(60),JPRIOR,KBEST,NCPASS
REAL COVAR(VARSZ2,NOSUB2),AVAR(NOFEAT,NOSUB),COV(VARSIZ),
*          AVEN(NOFEAT,NOSUB),R
DIMENSION CLSDFS(NOCLS),SURNO(NOCLS),SURDES(NOSUB),CLSDFS(NOCLS2),
*          SURNOS(NOCLS2),SURDS(NOSUB2),FLDSV(4,NOFLD2),
*          VERTX(2,TOTVT2),FLDSAV(4,NOFLD),VERTEX(2,TOTVT2)
*          ,DUMVEC(30)

REDUCE CLASS DESCRIPTION AND ARRAY CONTAINING NO OF SUBCLASSES
DO 150 I=1,NOCLS2
CLSDFS(I) = CLSDFS(I)
150 SURNOS(I) = SURNO(I)

REDUCE SUBCLASS DESCRIPTIONS
DO 160 I=1,NOSUB2
SURDS(I) = SURDES(I)
160

REDUCE FIELD INFORMATION
DO 170 I=1,NOFLD2
DO 170 J=1,4
170 FLDSV(J,I) = FLDSAV(J,I)

REDUCE VERTICES
DO 180 I=1,TOTVT2
DO 180 J=1,2
180 VERTX(J,I) = VERTEX(J,I)

ZERO OUT JUST PORTION OF COVAR THAT WILL CONTAIN SUBCLASSES
THAT HAVE BEEN GROUPED
DO 200 J=1,NOGRP
KR = GRPDEX(J) + 1
KF = KR + GROUPS(KR-1) - 1
IF (KR.GE. KF) GO TO 200
KK = SUPPTR(KR)
DO 195 LL = 1,VARSZ2
195 COVAR(LL,KK) = 0.0
200 CONTINUE

CHECK CLASSIFICATION CHANNELS AGAINST TRAINING CHANNELS
DO 220 J=1,NOFET2
DO 210 L=1,NOFEAT
IF (FETVC2(J).EQ. FETVEC(L)) GO TO 220
210 CONTINUE
WRITE(6,230) FETVC2(J),(FETVEC(K),K=1,NOFEAT)
230 FORMAT(' ** CHANNEL NO. ',I2,' IS NO. A TRAINING CHANNEL **',
*          ' 40X, TRAINING CHANNELS ARE - ',10X,30(I2,1X))

```

FILE: RENDAT

```

220 CALL EXIT
    DUMVEC(J) = L
    DO 100 JJ=1,NOSUR
    IS THIS SURCLASS A MEMBER OF SELECTED SURCLASSES
    IF (SURPTR(JJ) .LE. 0) HEAD(SAVTAP) DUMMY
    IF (SURPTR(JJ) .LE. 0) GO TO 100
    READ(SAVTAP)KEPPTS(JJ),COV,(AVAR(I,JJ),I=1,NOFEAT)

    C
    C
    C
    REDUCE BY CHANNELS
    NEWSUR = SURPTR(JJ)
    KK = 0
    DO 20 J=1,NOFET2
    K = DUMVEC(J)
    LOC = K*(K-1)/2
    DO 40 L=1,1
    KK = KK + 1
    WAT = LOC + DUMVEC(L)
    40 COV(KK) = COV(WAT)
    20 AVAR(J,JJ) = AVAR(K,JJ)

    C
    C
    C
    GROUP SURCLASSES
    IF (GRPCHK(JJ) .LE. 0) GO TO 60
    KK = 0
    DO 50 J1=1,NOFET2
    DO 50 J2=1,J1
    KK = KK + 1
    R = COVAR(KK,NEWSUR)
    COVAR(KK,NEWSUR) = R + (COV(KK)*(KEPPTS(JJ)-1)) +
    * (AVAR(J1,JJ)*AVAR(J2,JJ)*KEPPTS(JJ))
    50 CONTINUE
    GO TO 100
    60 DO 70 I=1,VARSZ2
    70 COVAR(I,NEWSUR) = COV(I)
    100 CONTINUE

    C
    C
    C
    GROUP MEANS
    DO 125 I=1,NOGRP
    KH = GRPDEX(I) + 1
    KE = KP + GROUPS(KH-1) - 1
    KR1 = GROUPS(KH)
    IF (KH .GE. KE) GO TO 130
    DO 120 J=1,NOFET2
    R = 0.0
    KPTS = 0
    DO 110 K=KR,KE
    KR2 = GROUPS(K)
    KPTS = KPTS + KEPPTS(KR2)
    110 R = R + AVAR(J,KR2) * KEPPTS(KR2)
    120 AVAR(J,KH1) = R / KPTS
    KEPPTS(KH1)=KPTS
    JJ = 0
    NEWSUR = SURPTR(KR1)
    DO 123 J1=1,NOFET2
    DO 123 J2=1,J1
    JJ = JJ + 1
    123 COVAR(JJ,NEWSUR) = (COVAR(JJ,NEWSUR) - (AVAR(J1,KH1)*
    * AVAR(J2,KH1))*KPTS) / (KPTS-1)
    125 CONTINUE
    REDUCE MEANS
    C
    130 DO 140 K=1,NOSUR2
    II = SURVCP(K)
    KEPPTS(K)=KEPPTS(II)
    DO 140 J=1,NOFET2
    140 AVAR(J,K) = AVAR(J,II)

    C
    C
    C
    RETURN
    END

```

RED00800
 RED00810
 RED00820
 RED00830
 RED00840
 RED00850
 RED00860
 RED00870
 RED00880
 RED00890
 RED00900
 RED00910
 RED00920
 RED00930
 RED00940
 RED00950
 RED00960
 RED00970
 RED00980
 RED00990
 RED01000
 RED01010
 RED01020
 RED01030
 RED01040
 RED01050
 RED01060
 RED01070
 RED01080
 RED01090
 RED01100
 RED01110
 RED01120
 RED01130
 RED01140
 RED01150
 RED01160
 RED01170
 RED01180
 RED01190
 RED01200
 RED01210
 RED01220
 RED01230
 RED01240
 RED01250
 RED01260
 RED01270
 RED01280
 RED01290
 RED01300
 RED01310
 RED01320
 RED01330
 RED01340
 RED01350
 RED01360
 RED01370
 RED01380
 RED01390
 RED01400
 RED01410
 RED01420
 RED01430
 RED01440
 RED01450
 RED01460
 RED01470
 RED01480
 RED01490
 RED01500
 RED01510

FILE: REDSAV

```

C      SURROUTINE REDSAV (ARRAY, TOP, BMFLG)
C
C      IMPLICIT INTEGER (A-Z)
C      DIMENSION ARRAY (1)
C
C      INCLUDE COMB1.LIST
C
C      INCLUDE COMB4.LIST
COMMON / INFORM / NOCLS2, NOSUR2, NOFET2, VARSZ2, TOTVT2, NOFLD2,
*      AVAR2, COVAR2, CLSID2, SURNO2, SURDS2, FLDSV2, VERTX2,
*      FETVC2 (30), SUHVC2 (75), SUHPTH (75), CLSVC2 (60),
*      KFPPTS (60), NOGRP, GMPNAM (60), GRPDEX (61),
*      GMPCHK (61), GROUPS (124)
COMMON / GLOBAL / HEAD (63), MAPTAP, DATAE, SAVTAP, BMFILE, BMKEY,
*      HISFIL, HISKEY, TRFORM, ERPTP, ERPKEY, MAPUNT, NOFILE,
*      DRUMAD, DRUMDS, PAGESIZ, DATFIL, STAFIL, ASAV, ASAVFL
*      NHSTUN, NHSTFI, SCTRIIN, MAPFIL
*      DOTUNT, DOTFIL, INCHPAS, TRNSFL, BMTRFL, HISTFL, PCHUNT,
*      CRDUNT, PRTUNT, RANDIO
CSEND
DIMENSION FETVEC (30)
REWIND SAVTAP
IF (STAFIL.EQ.-1) WRITE (6,100)
100 FORMAT (// T5, 'STAT FILE WAS NOT CREATED. EXITING FROM **REDSAV**')
IF (STAFIL.EQ.-1) CALL CMERR
IF (STAFIL.EQ. 0) GO TO 10
CALL FSHSFL (SAVTAP, STAFIL, ISTAT)
IF (ISTAT.EQ. 0) GO TO 10
FILNO = STAFIL + 1
WRITE (6,110) FILNO
110 FORMAT (// T5, 'ERROR IN POSITIONING STAT FILE TO FILE ', I3,
*      ' EXITING FROM REDSAV')
10 CONTINUE
HEAD (SAVTAP) NOCLS, NOSUB, NOFEAT, NOFLD, TOTVRT, (FETVEC (1), 1=1, NOFEAT)
C
C      COMPUTE BASES
C
VARSIZ = NOFEAT * (NOFEAT + 1) / 2
CLSID1 = 1
SUBNO1 = CLSID1 + NOCLS
SUBDS1 = SUBNO1 + NOCLS
FLDSV1 = SUBDS1 + NOSUB
VERTX1 = FLDSV1 + NOFLD * 4
C
CALL SAVFIL (ARRAY (FLDSV1), ARRAY (VERTX1), ARRAY (CLSID1),
*      ARRAY (SUBNO1), ARRAY (SUBDS1), NOFLD, NOCLS, NOSUB)
C
CALL CLSCHK (ARRAY (CLSID1), ARRAY (SUBDS1), ARRAY (FLDSV1),
*      ARRAY (VERTX1), ARRAY (SUBNO1), NOFEAT, FETVEC,
*      NOCLS, NOFLD, BMFLG, NOSUB)
C
C      COMPUTE REDUCED BASES
C
VARSZ2 = NOFET2 * (NOFET2 + 1) / 2
CLSID2 = 1
SURNO2 = CLSID2 + NOCLS2
SURDS2 = SURNO2 + NOCLS2
FLDSV2 = SURDS2 + NOSUR2
VERTX2 = FLDSV2 + NOFLD2 * 4
COVAR2 = VERTX2 + TOTVT2 * 2
AVAR2 = COVAR2 + VARSZ2 * NOSUR2
COV1 = AVAR2 + (NOFEAT * NOSUB)
TIPTOP = COV1 + VARSIZ
RANDCOR = TOP - TIPTOP
IF (RANDCOR.LT. 0) GO TO 50
C
CALL REDDAT (ARRAY (COVAR2), ARRAY (AVAR2), ARRAY (CLSID1), ARRAY (SURNO1),
*      ARRAY (SUBDS1), ARRAY (FLDSV1), ARRAY (VERTX1),
*      ARRAY (COV1), ARRAY (AVAR2), ARRAY (CLSID2),
*      ARRAY (SURNO2), ARRAY (SURDS2), ARRAY (FLDSV2),
*      ARRAY (VERTX2), NOFEAT, VARSIZ, NOCLS, NOFLD,
*      NOSUB, FETVEC)
C
GO TO 70
50 WRITE (6,60) NOFET2, NOSUR2, NOCLS2
60 FORMAT (// T5, 'AS REQUESTED ', I2, ' CHANNELS, ', I2, ' SUBCLASSES, ', I2, ' AND ', I2, ' CLASSES. // THIS COMBINATION OF STATS WILL NOT FIT IN RED00780

```

FILE: REDSAV

*CONF. PLEASE REDUCE REQUEST.**)
CALL CMERR

C 70 CONTINUE
RETURN
END

RED00800
RED00810
RED00820
RED00830
RED00840
RED00850

~~19-76~~

449

FILE: PREAD

SUBROUTINE PREAD(BEGADD, /WHERE/, TOTWDS, STATUS)

THIS SUBROUTINE SIMULATES THE RANDOM ACCESS READ OF A
WORKFILE USED TO STORE PROGRAM DATA TEMPORARILY DURING A
LARSYS PROCESSOR RUN. THE CALLING ARGUMENTS ARE:

BEGADD - THE NUMBER OF WORDS FROM THE BEGINNING OF THE FILE
WHERE THE READ IS TO BEGIN.
WHEREF - WHERE THE DATA READ IS TO BE PUT (OUTPUT AREA).
TOTWDS - THE TOTAL NUMBER OF WORDS TO BE READ.
STATUS - SET TO ZERO WHEN I/O IS COMPLETE (NO LONGER USED,
BUT MUST BE RETURNED AS 0).

IMPLICIT INTEGER (A-Z)

HANSEN / VERSION 0800/R/31/77

DIMENSION BUFFER(200), WHERE(1)
BUFSIZ=200

BUFFER AND BUFSIZ ARE SET TO THE MOST EFFICIENT SIZE TO
MATCH THE PHYSICAL RECORD SIZE OF THE I/O BUFFER. IN
CMS IT IS 800 BYTES - 200 WORDS.

STATUS=0
LUD=22

LUD IS THE LOGICAL UNIT NUMBER WHERE THE FORTRAN DIRECT
ACCESS FILE IS STORED.

I1=1
J1=MOD(BEGADD, BUFSIZ)
IF (J1.EQ.0) J1=BUFSIZ

J1 IS THE RELATIVE ADDRESS OF THE BEGINNING WORD IN THE
FIRST RECORD TO BE READ. IF IT IS 0, IT IS THE LAST WORD
IN THE RECORD.

J2=BEGADD
J3=BEGADD + TOTWDS - 1

J2 AND J3 ARE THE BEGINNING AND ENDING WORDS OF THE DATA
TO BE READ.

J4=MOD(J3, BUFSIZ)
IF (J4.EQ.0) J4=BUFSIZ

J4 IS THE RELATIVE ADDRESS OF THE ENDING WORD IN THE FINAL
RECORD TO BE READ. IF IT IS 0, IT IS THE LAST WORD IN
THE RECORD.

REGREC=((J2-1)/BUFSIZ) + 1
ENDREC=((J3-1)/BUFSIZ) + 1
IF (REGREC.EQ.ENDREC) GOTO 300

REGREC AND ENDREC ARE THE RELATIVE ADDRESSES (RECORD
NUMBERS) OF THE FIRST AND LAST RECORDS TO BE READ. IF
THEY ARE EQUAL THEN WE ARE TO BEGIN AND END IN THE
SAME RECORD.

K1=REGREC

READ THE FIRST RECORD AND MOVE THE REQUIRED PORTION TO
THE OUTPUT AREA.

READ(LUD, K1) BUFFER
DO 200 K2=J1, BUFSIZ
WHERE(I1)=BUFFER(K2)
I1=I1+1
CONTINUE
K1=K1+1

READ IN THE NEXT RECORD. IF IT IS THE FINAL RECORD TO
BE READ, GO TO THE FINAL RECORD MOVE CODE ELSE MOVE THE
ENTIRE RECORD TO THE OUTPUT AREA.

IF (K1.EQ.ENDREC) GOTO 230

RRE00010
RRE00020
RRE00030
RRE00040
RRE00050
RRE00060
RRE00070
RRE00080
RRE00090
RRE00100
RRE00110
RRE00120
RRE00130
RRE00140
RRE00150
RRE00160
RRE00170
RRE00180
RRE00190
RRE00200
RRE00210
RRE00220
RRE00230
RRE00240
RRE00250
RRE00260
RRE00270
RRE00280
RRE00290
RRE00300
RRE00310
RRE00320
RRE00330
RRE00340
RRE00350
RRE00360
RRE00370
RRE00380
RRE00390
RRE00400
RRE00410
RRE00420
RRE00430
RRE00440
RRE00450
RRE00460
RRE00470
RRE00480
RRE00490
RRE00500
RRE00510
RRE00520
RRE00530
RRE00540
RRE00550
RRE00560
RRE00570
RRE00580
RRE00590
RRE00600
RRE00610
RRE00620
RRE00630
RRE00640
RRE00650
RRE00660
RRE00670
RRE00680
RRE00690
RRE00700
RRE00710
RRE00720
RRE00730
RRE00740
RRE00750
RRE00760
RRE00770
RRE00780
RRE00790

FILE: RREAD

```

      READ(LUD,K1) BUFFER
      DO 220 K2=1,RIIFSIZ
      WHERE(I1)=BUFFER(K2)
      I1=I1+1
220   CONTINUE
      GOTO 210
C
C
C
      READ THE FINAL RECORD. MOVE THE REQUIRED PORTION TO THE
      OUTPUT AREA AND RETURN.
230   READ(LUD,K1) BUFFER
      DO 240 K2=1,J4
      WHERE(I1)=BUFFER(K2)
      I1=I1+1
240   CONTINUE
      RETURN
C
C
C
      HERE WE BEGIN AND END IN THE SAME RECORD. THEREFORE WE
      ONLY MOVE THE REQUIRED PORTION OF THE DATA TO THE OUTPUT
      AREA AND RETURN.
300   READ(LUD,HEGPEC) BUFFER
      DO 310 K2=J1,J4
      WHERE(I1)=BUFFER(K2)
      I1=I1+1
310   CONTINUE
      RETURN
      END

```

```

RRE00800
RRE00810
RRE00820
RRE00830
RRE00840
RRE00850
RRE00860
RRE00870
RRE00880
RRE00890
RRE00900
RRE00910
RRE00920
RRE00930
RRE00940
RRE00950
RRE00960
RRE00970
RRE00980
RRE00990
RRE01000
RRE01010
RRE01020
RRE01030
RRE01040
RRE01050
RRE01060
RRE01070

```

FILE: RWRITE

SUBROUTINE RWRITE(BEGADD,WHERE,TOTWDS,STATUS)

THIS SUBROUTINE SIMULATES THE RANDOM ACCESS WRITE OF A
WORKFILE USED TO STORE PROGRAM DATA TEMPORARILY DURING
A LARSYS PROCESSOR RUN. THE CALLING ARGUMENTS ARE:

BEGADD - THE NUMBER OF WORDS FROM THE BEGINNING OF THE
FILE WHERE THE WRITE IS TO BEGIN.
WHERE - WHERE THE DATA TO BE WRITTEN IS STORED (INPUT
AREA).
TOTWDS - THE TOTAL NUMBER OF WORDS TO BE WRITTEN.
STATUS - SET TO ZERO WHEN I/O IS COMPLETE (NO LONGER
USED, BUT MUST BE RETURNED AS 0)

IMPLICIT INTEGER (A-Z)

HANSEN / VERSION 0800/8/31/77

DIMENSION BUFFER(200),WHEPE(1)
RUFISZ=200

BUFFER AND RUFISZ ARE SET TO THE MOST EFFICIENT SIZE TO
MATCH THE PHYSICAL RECORD SIZE OF THE I/O BUFFER. IN
CMS IT IS 800 BYTES - 200 WORDS.

STATUS=0
LUD=22

LUD IS THE LOGICAL UNIT NUMBER WHERE THE FORTRAN DIRECT
ACCESS FILE IS STORED.

I1=1
J1=MOD(BEGADD,RUFISZ)
IF(J1.EQ.0) J1=RUFISZ

J1 IS THE RELATIVE ADDRESS OF THE BEGINNING WORD IN THE
FIRST RECORD TO BE WRITTEN. IF IT IS 0, IT IS THE LAST
WORD IN THE RECORD.

J2=BEGADD
J3=BEGADD + TOTWDS - 1

J2 AND J3 ARE THE BEGINNING AND ENDING WORDS OF THE DATA
TO BE WRITTEN.

J4=MOD(J3,RUFISZ)
IF(J4.EQ.0) J4=RUFISZ

J4 IS THE RELATIVE ADDRESS OF THE ENDING WORD IN THE
FINAL RECORD TO BE WRITTEN. IF IT IS 0, IT IS THE
LAST WORD IN THE RECORD.

REGREC=((J2-1)/RUFISZ) + 1
ENDREC=((J3-1)/RUFISZ) + 1
IF(REGREC.EQ.ENDREC) GOTO 300

REGREC AND ENDREC ARE THE RELATIVE ADDRESSES (RECORD
NUMBERS) OF THE FIRST AND LAST RECORDS TO BE WRITTEN.
IF THEY ARE EQUAL, THEN WE ARE TO BEGIN AND END IN
THE SAME RECORD.

K1=REGREC

READ THE FIRST RECORD (THERE MIGHT BE DATA IN THE PORTION
OF THE RECORD WE ARE NOT WRITING). MOVE THE REQUIRED
PORTION OF THE DATA FROM THE INPUT AREA TO THE BUFFER
AND WRITE IT OUT.

READ(LUD,K1) BUFFER
DO 200 K2=J1,RUFISZ
BUFFER(K2)=WHEPE(I1)
I1=I1+1
200 CONTINUE
WRITE(LUD,K1) BUFFER

PUMP THE RECORD COUNTER AND CHECK TO SEE WHETHER WE
ARE AT THE FINAL RECORD TO BE WRITTEN. IF WE ARE, GO

RRE00010
RRE00020
RRE00030
RRE00040
RRE00050
RRE00060
RRE00070
RRE00080
RRE00090
RRE00100
RRE00110
RRE00120
RRE00130
RRE00140
RRE00150
RRE00160
RRE00170
RRE00180
RRE00190
RRE00200
RRE00210
RRE00220
RRE00230
RRE00240
RRE00250
RRE00260
RRE00270
RRE00280
RRE00290
RRE00300
RRE00310
RRE00320
RRE00330
RRE00340
RRE00350
RRE00360
RRE00370
RRE00380
RRE00390
RRE00400
RRE00410
RRE00420
RRE00430
RRE00440
RRE00450
RRE00460
RRE00470
RRE00480
RRE00490
RRE00500
RRE00510
RRE00520
RRE00530
RRE00540
RRE00550
RRE00560
RRE00570
RRE00580
RRE00590
RRE00600
RRE00610
RRE00620
RRE00630
RRE00640
RRE00650
RRE00660
RRE00670
RRE00680
RRE00690
RRE00700
RRE00710
RRE00720
RRE00730
RRE00740
RRE00750
RRE00760
RRE00770
RRE00780
RRE00790

FILE: RWRITE

```

C      TO THE FINAL RECORD WRITE CODE ELSE MOVE AND WRITE
C      THE ENTIRE RECORD.
C
210    K1=K1+1
      IF (K1.EQ.ENDREC) GOTO 230
      DO 220 K2=1,RUFSIZ
      BUFFER(K2)=WHERE(I1)
      I1=I1+1
220    CONTINUE
      WRITE(LUD,K1) BUFFER
      GOTO 210
C
C      READ THE FINAL RECORD. MOVE THE REQUIRED PORTION OF THE
C      DATA FROM THE INPUT AREA TO THE BUFFER, WRITE IT OUT
C      AND RETURN.
C
230    READ(LUD,K1) BUFFER
      DO 240 K2=1,J4
      BUFFER(K2)=WHERE(I1)
      I1=I1+1
240    CONTINUE
      WRITE(LUD,K1) BUFFER
      RETURN
C
C      HERE WE BEGIN AND END IN THE SAME RECORD. THEREFORE WE
C      ONLY MOVE THE REQUIRED PORTION OF THE DATA FROM THE
C      INPUT AREA TO THE BUFFER, WRITE IT OUT AND RETURN.
C
300    READ(LUD,BEGREC) BUFFER
      DO 310 K2=J1,J4
      BUFFER(K2)=WHERE(I1)
      I1=I1+1
310    CONTINUE
      WRITE(LUD,BEGREC) BUFFER
      RETURN
      END

```

WRE00800
 WRE00810
 WRE00820
 WRE00830
 WRE00840
 WRE00850
 WRE00860
 WRE00870
 WRE00880
 WRE00890
 WRE00900
 WRE00910
 WRE00920
 WRE00930
 WRE00940
 WRE00950
 WRE00960
 WRE00970
 WRE00980
 WRE00990
 WRE01000
 WRE01010
 WRE01020
 WRE01030
 WRE01040
 WRE01050
 WRE01060
 WRE01070
 WRE01080
 WRE01090
 WRE01100
 WRE01110
 WRE01120
 WRE01130
 WRE01140
 WRE01150

FILE: SAVFIL

| | | |
|-------|---|----------|
| | SUBROUTINE SAVFIL (FLDSAV, VERTEX, CLSID, SUBNO, SUBDES,
NOFLD, NOCLS, NOSUB) | SAV00010 |
| C | IMPLICIT INTEGER (A-Z) | SAV00020 |
| C | INCLUDE COMHKL.LIST | SAV00030 |
| C | INCLUDE COMHKL.LIST | SAV00040 |
| | COMMON/INFORM/NOCL52, NOSUR2, NOFET2, VARS22, TOTVT2, NOFLD2,
AVAR2, COVAR2, CLSID2, SUBNO2, SUBNS2, FLDSV2, VERTX2,
FETVC2(30), SUHVC2(75), SURPTR(75), CLSVC2(60),
KEPPTS(60), NOGRP, GRPNAM(60), GRPDEX(61),
GRPCHK(61), GROUPS(124) | SAV00050 |
| | COMMON/GLOBAL/HEAD(63), MAPTAP, DATAPE, SAVTAP, BMFILE, RMKEY,
HISFIL, HISKEY, TRFORM, ERPTP, ERPKY, MAPUNT, NOFILE,
DRUMAD, DRMMUS, PAGESZ, DATFIL, STAFIL, ASAV, ASAVFL,
NHSTON, NHSTFI, SCTRUN, MAPFIL,
DOTUNT, DOTFIL, NCHPAS, TRNSFL, BMTRFL, HISTFL, PCHUNT,
CRDUNT, PRUNT, RANDIO | COM00010 |
| CSFND | DIMENSION FLDSAV(4, NOFLD), VERTEX(1), CLSID(1), SUBNO(1),
SUBDES(1) | COM00020 |
| | IR=0 | COM00030 |
| | DO 40 J=1, NOFLD | COM00040 |
| | READ(SAVTAP) (FLDSAV(I, J), I=1, 4) | COM00050 |
| | FLD = 2 * FLDSAV(4, J) | COM00060 |
| | READ(SAVTAP) (VERTEX(I+IR), I=1, FLD) | COM00070 |
| | IR = IR + FLD | COM00080 |
| 40 | CONTINUE | COM00090 |
| | READ(SAVTAP) (CLSID(I), I=1, NOCLS), (SUBNO(I), I=1, NOCLS),
(SUBDES(I), I=1, NOSUB) | COM00100 |
| | RETURN | COM00110 |
| C | END | SAV00000 |

FILE: SEARCH

```

SUBROUTINE SEARCH(*,*,ENDTAP,IRUF,IRPDS,NDSPR)
IMPLICIT INTEGER (A-Z)
C*
C* INTERNAL ROUTINE TO SEARCH FOR CORRECT SCAN LINE
C*
LOGICAL * LSCAN(4), ISCAN(4), KSCAN(4)
DIMENSION IJUF(765), KDUM(1), LDUM(1)
COMMON /TAPERD/ IUNIT, IFRST, FSCAN, SAMEND, SAMINC, READY, NSCAN,
* LINC, IO(200), DSL, LHIIF(30), JHEC(30), IHYE(30), NHUF, FILENO, LINEND,
* LINC, NSAMP, NOCHAN, FORMT
EQUIVALENCE (ISCAN, SCAN), (KDUM(1), KSCAN(1)), (LDUM(1), LSCAN(1))
SCAN=0
WRITE (A,600) FSCAN
600 FORMAT(' SEARCHING FOR LINE',I5)
WRITE (A,610) IRPDS, NDSPR
610 FORMAT(' RECORDS PER SCAN',I5,' SCANS PER RECORD',I5)
IRACK=-5
IF (IRPDS.GT.5) IRACK=-NRPDS
IF (FSCAN.LE.5) IRACK=-NRPDS
RSKIP=-IRACK
DO 620 I=1,RSKIP
620 BACKSPACE IUNIT
ITRY=1
5 READ (IUNIT,910,END=630) (IRUF(I),I=1,16)
910 FORMAT(1H44)
GO TO 640
630 ENDTAP = -1
RETURN 1
640 CONTINUE
KDUM(1)=IRUF(1)
LDUM(1)=IRUF(16)
IF (FORMAT.EQ.1) ISCAN(3) = LSCAN(3)
IF (FORMAT.EQ.1) ISCAN(4) = LSCAN(4)
IF (FORMAT.EQ.2) ISCAN(3) = KSCAN(1)
IF (FORMAT.EQ.2) ISCAN(4) = KSCAN(2)
WRITE (A,650) SCAN
650 FORMAT(' SCAN NO',I5,'X')
IF (SCAN.EQ.FSCAN) GO TO 10
IF (SCAN.EQ.FSCAN+NDSPR) GO TO 30
IF (ITRY.EQ.10) GO TO 20
6 CONTINUE
ITRY=ITRY+1
GO TO 5
10 WRITE (A,700) ITRY
700 FORMAT(' FOUND IT AFTER',I3,' TRIES')
DO 15 I=1,IRPDS
15 BACKSPACE IUNIT
RETURN
20 IF (ITRY.LT.2*NRPDS) GO TO 6
WRITE (A,800) ITRY
800 FORMAT(' FAILED AFTER',I5,' TRIES--ABORTING')
CALL CMERR
30 CONTINUE
WRITE (A,900) FSCAN
900 FORMAT(' SCAN',I5,' IS MISSING--USING PREVIOUS SCAN INSTEAD')
BACKSPACE IUNIT
RETURN 2
END

```

SEA00010
SEA00020
SEA00030
SEA00040
SEA00050
SEA00060
SEA00070
SEA00080
SEA00090
SEA00100
SEA00110
SEA00120
SEA00130
SEA00140
SEA00150
SEA00160
SEA00170
SEA00180
SEA00190
SEA00200
SEA00210
SEA00220
SEA00230
SEA00240
SEA00250
SEA00260
SEA00270
SEA00280
SEA00290
SEA00300
SEA00310
SEA00320
SEA00330
SEA00340
SEA00350
SEA00360
SEA00370
SEA00380
SEA00390
SEA00400
SEA00410
SEA00420
SEA00430
SEA00440
SEA00450
SEA00460
SEA00470
SEA00480
SEA00490
SEA00500
SEA00510
SEA00520
SEA00530
SEA00540
SEA00550
SEA00560
SEA00570
SEA00580

FILE: SETMRG

SUBROUTINE SETMRG(A,B,C)
IMPLICIT INTEGER (A-C)
RETURN
END

SET00010
SET00020
SET00030
SET00040

ORIGINAL PAGE IS
OF POOR QUALITY

```

SUBROUTINE SETUP7 (ARRAY,TOP,ITIME)
IMPLICIT INTEGER (A-X)
*****
SETUP7 READS AND ANALYZES ALL CONTROL CARD INPUT FOR THE
ISOCLS PROCESSOR

INCLUDE COMBK4.LIST
INCLUDE COMBK5.LIST
INCLUDE COMBK6.LIST
INCLUDE COMBK16.LIST
DIMENSION HED1(15),HED2(15),DATE(3),COMENT(15)
EQUIVALENCE (HED1(1),HEAD(4)),(DATE(1),HEAD(22)),
              (HED2(1),HEAD(30)),(COMENT(1),HEAD(48))
COMMON/PASS/STOP,LNCAT,NMIN,KPN,STDMAX,DLMIN,SEP,
              MAP,SPTRIG,IHO,KPTS,NOPTS,PUNCH,
              ICHN,CHNTHS,ICHAIN(62),NWDS,BEGIN,BEGIN1,
              BEGIN2,BEGIN3,CLSNAM,NOFLD,IPT,TOTWRD,TOTPTS,
              NCLASS,NOCLS,TOTSUR,TOTFLD,TOTVRT,NOCL,NVRT,
              ,NXTCLS,NOFFAT,MAXCLS,FEFVFC(30),SYMMTX(62)
              ,VARSTZ,STATKY,ISOKEY,MAPFMT,MAPKEY,SEQUEN(20),PERCEN,SIMERP
              ,IORDER,INUNIT,INFILE,INITM,PMIN,SUBVEC(62),NOSUB2,CHNVLC(30)
              ,NOCHAN,ERCOMP,NOSEQ,MEANDU,MEANDU,
              ,SYMDU,SYMDU,ITRIGO,ITRIGU,DOLAG,
              ,DUFLAG,MODU,SDOTS(60),NSDOTS,SUNCOR(30),LLNCAT,
              ,DVET(250,2),DRECT(60,2),DVPNT(11,2),IDCNT(2),NDUU(2)
              ,MXFET1,MXPDP
REAL SUNCOR
COMMON/GLOBAL/HEAD(63),MAPTAP,DATAP,SAVTAP,BMFILE,BMKEY,
              HISFIL,HISKEY,TRFORM,ERIPT,ERPKEY,MAPUNT,NOFILE,
              ,DRUMAD,DRMWDS,PAGSZ,DATFIL,STAFIL,ASAV,ASAVL
              ,NHSTUN,NHSTFI,SCRPUN,MAPFIL
              ,DOTUNT,DOTFIL,NCHPAS,TRNSFL,BMTRFL,HISTFL,PCHUNT,
              ,CRDUNT,PRUNT,RANDIO
COMMON/ISOLNK/SUNANG(8),ISUNT,ISUNC,SMSTR,SMSTP,SMINC,LINKP
*****
REAL DLMIN,CHNTHS,SEP,STDMAX,PERCEN
DIMENSION ARRAY(1),CHAR(2)
DIMENSION COMVEC(2),EQUVEC(2)
DATA COMVEC/1.,1./,EQUVEC/1.,1./
DATA MINUS/'-'/
EQUIVALENCE (STOP,ISTOP)
DIMENSION INVEC(30),CARD(62),ACARD(20),SLASH(2)
DATA SLASH/1.,1./
DATA INVEC /'CHAN','ISTO','NMIN','KRN','STDM','DLMI',
             ,'SEP','HED1','HED2','DATE','END','COMM',
             ,'SYMB','MEAN','MAP','OPTI','CLAS','CHA1',
             ,'CLUS','FORM','DATA','STAT','SECU','PERC',
             ,'MODU','SUCC','PMIN','DOTF','SUNA','DOTS'/
DIMENSION SMRLS(60)
DATA IRCD/'1'/'U'/'LBCD'/'L'/'MBCD'/'M'/'
DATA CHAR/'1.'/'='/'EQUAL'/'='/'
DATA PRCD/'P'/'FB'/'E'/'DBCD'/'D'/'
DATA FRCD/'F'/'CB'/'C'/'
DATA TRCD/'T'/'SB'/'S'/'OB'/'O'/'BLANK'/' '/'COMMA'/'.'/
DATA SMRLS /'1'/'2'/'3'/'4'/'5'/'6'/'7'/'8'/'9'/'A'/'R'/'C'/'D',
            'I'/'F'/'G'/'H'/'J'/'L'/'M'/'N'/'O'/'P'/'Q',
            'R'/'S'/'T'/'U'/'V'/'W'/'X'/'Y'/'Z'/'_'/'-',
            '!'/'@'/'#'/'$'/'%'/'&'/'*'/'+'/'='/'>/'?'/'\H'/'.'/
IF(ITIME.NE.1)GO TO 5
SET UP DEFAULT VALUES FOR INPUT PARAMETERS

ERCOMP=0
IORDER=0
NOSEQ = 2
NOCHAN = 0
SEP=1
NOSUB2=0
PMIN=1
INITM=0
MAPKEY=1
MAPFMT = 0
ICNT=0
PINCH=0
DLMIN=3.2

```


FILE: SETUP7

```

STDMAK=4.5
MAP=20
STATKY=0
ISTOP=10
NMIN=30
KRN=20
NCLASS=1
MAXCLS = 60
ICHN=0
PERCEN = .2
IPCT = 80
ISUNC=0
ISUNT=0
NSDOTS=0
DO 1 I=1,MAXPOP
1 SYMPTX(I) = SMBLS(I)
NOFEAT = 0
ISTART = 0
DO 2 I=1,19,2
2 SEQUEN(I)=SRCD
DO 3 I=2,18,2
3 SEQUEN(I)=CRCD
SFQUEN(20)=0
5 CONTINUE
IF (ITIME,NE,1) WRITE(6,HEAD)
WRITE(6,630)
C
C PUT THE NEXT CARD IN THE REREAD BUFFER
1000 RRUNIT=30
1000 READ(21,1000) (ACARD(I),I=1,20)
1000 FORMAT(20A4)
1000 WRITE(RRUNIT,1000) (ACARD(I),I=1,20)
1000 REWIND RRUNIT
C
C READ IN CARD
C
C READ(30,480) CODE,CARD
C REWIND RRUNIT
C WRITE(6,550) CODE,CARD
C COL = 0
C DO 20 I=1,30
C IF (CODE.EQ.INVEC(I)) GO TO
C * (30,50,70,80,90,100,110,130,140,150,280,160,
C * 170,250,210,240,200,235,230,220,260,270,246,249,
C * 256,283,284,300,340,380),I
20 CONTINUE
WRITE(6,490) CODE,CARD
GO TO 10
C CHANNEL CARD
30 J=NXTCHR(CARD,COL)
IF (J.EQ.BLANK) GO TO 10
IF (ITIME,FO,1) GO TO 35
WRITE(6,640)
GO TO 10
35 CONTINUE
IF (J.EQ.SRCD) GO TO 37
IF (J.EQ.DRCD) GO TO 43
36 WRITE(6,645)
645 FORMAT(' ERROR ON CHANNEL CARD')
GO TO 10
37 M=FINDI2(CARD,COL,EQUVEC)
IF (M.EQ.-1) GO TO 36
NOCHAN=NUMBER(CARD,COL,CHNVC,NOCHAN)
COL = COL-1
CALL ORDER(CHNVC,NOCHAN)
GO TO 30
43 M=FINDI2(CARD,COL,EQUVEC)
IF (M.EQ.-1) GO TO 36
NOFEAT = NUMBER(CARD,COL,FETVEC,NOFEAT)
COL = COL-1
CALL ORDER(FETVEC,NOFEAT)
GO TO 30
C*
C* ISTOP CARD (MAXIMUM NUMBER OF ITERATIONS)
C*
50 J = NXTCHR(CARD,COL)
IF (J.EQ.BLANK) GO TO 10

```

SET00800
 SET00810
 SET00820
 SET00830
 SET00840
 SET00850
 SET00860
 SET00870
 SET00880
 SET00890
 SET00900
 SET00910
 SET00920
 SET00930
 SET00940
 SET00950
 SET00960
 SET00970
 SET00980
 SET00990
 SET01000
 SET01010
 SET01020
 SET01030
 SET01040
 SET01050
 SET01060
 SET01070
 SET01080
 SET01090
 SET01100
 SET01110
 SET01120
 SET01130
 SET01140
 SET01150
 SET01160
 SET01170
 SET01180
 SET01190
 SET01200
 SET01210
 SET01220
 SET01230
 SET01240
 SET01250
 SET01260
 SET01270
 SET01280
 SET01290
 SET01300
 SET01310
 SET01320
 SET01330
 SET01340
 SET01350
 SET01360
 SET01370
 SET01380
 SET01390
 SET01400
 SET01410
 SET01420
 SET01430
 SET01440
 SET01450
 SET01460
 SET01470
 SET01480
 SET01490
 SET01500
 SET01510
 SET01520
 SET01530
 SET01540
 SET01550
 SET01560
 SET01570
 SET01580

FILE: SETUP7

| | | |
|-----|--|----------|
| | COL=COL-1 | SET01590 |
| | J = NUMBER(CARD,COL,ISTOP,ISTART) | SET01600 |
| | GO TO 10 | SET01610 |
| C* | | SET01620 |
| C* | NMIN CARD (MINIMUM NUMBER OF POINTS PER CLUSTER) | SET01630 |
| C* | | SET01640 |
| 70 | J = NXTCHR(CARD,COL) | SET01650 |
| | IF (J.EQ.BLANK) GO TO 10 | SET01660 |
| | COL = COL-1 | SET01670 |
| | J = NUMBER(CARD,COL,NMIN,ISTART) | SET01680 |
| | GO TO 10 | SET01690 |
| C* | | SET01700 |
| C* | KRN CARD (NUMBER OF ITERATIONS PER FULL OUTPUT) | SET01710 |
| C* | | SET01720 |
| 80 | J = NXTCHR(CARD,COL) | SET01730 |
| | IF (J.EQ.BLANK) GO TO 10 | SET01740 |
| | COL = COL-1 | SET01750 |
| | J = NUMBER(CARD,COL,KRN,ISTART) | SET01760 |
| | GO TO 10 | SET01770 |
| C* | | SET01780 |
| C* | STDMAX CARD (MAXIMUM STANDARD DEVIATION PER CLUSTER) | SET01790 |
| C* | | SET01800 |
| 90 | J = FLTNUM(CARD,COL,STDMAX,1) | SET01810 |
| | GO TO 10 | SET01820 |
| C* | | SET01830 |
| C* | DLMIN CARD (MINIMUM DISTANCE BETWEEN CLUSTER MEANS) | SET01840 |
| C* | | SET01850 |
| 100 | J = FLTNUM(CARD,COL,DLMIN,1) | SET01860 |
| | GO TO 10 | SET01870 |
| C* | | SET01880 |
| C* | SEP CARD (DISTANCE FOR SPLITTING) | SET01890 |
| C* | | SET01900 |
| 110 | J = FLTNUM(CARD,COL,SEP,1) | SET01910 |
| | SPTRIG=1 | SET01920 |
| | GO TO 10 | SET01930 |
| C* | | SET01940 |
| C* | HED1 CARD | SET01950 |
| C* | | SET01960 |
| 130 | READ (30,500)HED1 | SET01970 |
| | REWIND RRUNIT | SET01980 |
| | GO TO 10 | SET01990 |
| C* | | SET02000 |
| C* | HED2,CARD | SET02010 |
| C* | | SET02020 |
| 140 | READ (30,500)HED2 | SET02030 |
| | REWIND RRUNIT | SET02040 |
| | GO TO 10 | SET02050 |
| C* | | SET02060 |
| C* | DATE CARD | SET02070 |
| C* | | SET02080 |
| 150 | READ (30,510) DATE | SET02090 |
| | REWIND RRUNIT | SET02100 |
| | GO TO 10 | SET02110 |
| C* | | SET02120 |
| C* | COMMENT CARD | SET02130 |
| C* | | SET02140 |
| 160 | READ (30,500)COMENT | SET02150 |
| | REWIND RRUNIT | SET02160 |
| | GO TO 10 | SET02170 |
| C* | | SET02180 |
| C* | SYMBOLS CARD | SET02190 |
| C* | | SET02200 |
| 170 | CONTINUE | SET02210 |
| 180 | ICNT=ICNT + 1 | SET02220 |
| | IF (ICNT.GT.MAXPOP) GO TO 10 | SET02230 |
| | SYMMTX(ICNT)=BLANK | SET02240 |
| 190 | M=NXTCHR(CARD,COL) | SET02250 |
| | IF (M.EQ.BLANK) GO TO 10 | SET02260 |
| | IF (M.EQ.COMMA) GO TO 180 | SET02270 |
| | SYMMTX(ICNT) = M | SET02280 |
| 195 | M=NXTCHR(CARD,COL) | SET02290 |
| | IF (M.EQ.BLANK) GO TO 10 | SET02300 |
| | IF (M.EQ.COMMA) GO TO 195 | SET02310 |
| | GO TO 180 | SET02320 |
| C* | | SET02330 |
| C* | MAXCLASS CARD (NO. CLASSES FOR THIS EXECUTION OF ISOCIS- STATISTIC | SET02340 |
| C* | FILE WILL BE WRITTEN AFTER 'NCLASS' CLASSES HAVE | SET02350 |
| C* | BEEN CLUSTERED) | SET02360 |
| C* | | SET02370 |

FILE: SETUP7

| | | |
|-----|--|----------|
| 200 | IF (ITIME.EQ.1) GO TO 205 | SET02380 |
| | WRITE(6,650) | SET02390 |
| | GO TO 10 | SET02400 |
| 205 | J=NUMBER(CARD,COL,NCLASS,ISTART) | SET02410 |
| | GO TO 10 | SET02420 |
| C* | | SET02430 |
| C* | MAP CARD (NUMBER OF ITERATIONS TO OUTPUT MAP) | SET02440 |
| C* | | SET02450 |
| 210 | J=NXTCHR(CARD,COL) | SET02460 |
| | IF (J.EQ.BLANK) GO TO 10 | SET02470 |
| | COL=COL-1 | SET02480 |
| | J = NUMBER(CARD,COL,MAP ,ISTART) | SET02490 |
| | GO TO 10 | SET02500 |
| C* | | SET02510 |
| C* | PUNCH CARD (PUNCH STATISTICS ON CARDS) | SET02520 |
| C* | | SET02530 |
| 215 | M = FIND12(CARD,COL,CHAR) | SET02540 |
| | IF (M.NE.2) PUNCH = 1 | SET02550 |
| | IF (M.NE.2) GO TO 245 | SET02560 |
| | J = NUMBER(CARD,COL,PUNCH ,ISTART) | SET02570 |
| | GO TO 245 | SET02580 |
| C* | | SET02590 |
| C* | MAP FORMAT CARD | SET02600 |
| C* | | SET02610 |
| 220 | M = NXTCHR(CARD,COL) | SET02620 |
| | IF (M.EQ.URCD) MAPFMT = 1 | SET02630 |
| | IF (M.EQ.LRCD) MAPFMT = 2 | SET02640 |
| | IF (M.EQ.BLANK) MAPFMT=1 | SET02650 |
| | GO TO 10 | SET02660 |
| C* | | SET02670 |
| C* | CLUSTERS CARD (MAX. NO. OF CLUSTERS PER CLASS) | SET02680 |
| C* | | SET02690 |
| 230 | J=NXTCHR(CARD,COL) | SET02700 |
| | IF (J.EQ.BLANK) GO TO 10 | SET02710 |
| | COL=COL-1 | SET02720 |
| | J = NUMBER(CARD,COL,MAXCLS,ISTART) | SET02730 |
| | GO TO 10 | SET02740 |
| C* | | SET02750 |
| C* | CHAIN CARD (CHAIN CLUSTERS WHICH ARE DLIN UNITS APART) | SET02760 |
| 235 | ICHN=1 | SET02770 |
| | J=FLINUM(CARD,COL,CHNTHS,1) | SET02780 |
| | GO TO 10 | SET02790 |
| C | | SET02800 |
| C | OPTION CARD | SET02810 |
| C | | SET02820 |
| 240 | J = NXTCHR(CARD,COL) | SET02830 |
| | IF (J.EQ.BLANK) GO TO 10 | SET02840 |
| C | | SET02850 |
| C | ORDER COLOR KEYS | SET02860 |
| C | | SET02870 |
| | IF (J.EQ.ORCD) IORDER = 1 | SET02880 |
| C | | SET02890 |
| C | ERROR COMPUTATION | SET02900 |
| C | | SET02910 |
| | IF (J.NE.ERCD) GO TO 241 | SET02920 |
| | COL=COL+1 | SET02930 |
| | J=NXTCHR(CARD,COL) | SET02940 |
| | IF (J.EQ.CRCD) ERCOMP=1 | SET02950 |
| C | | SET02960 |
| | GO TO 245 | SET02970 |
| 241 | CONTINUE | SET02980 |
| C | | SET02990 |
| C | PUNCH CARD | SET03000 |
| C | | SET03010 |
| | IF (J.EQ.PRCD) GO TO 215 | SET03020 |
| C* | | SET03030 |
| C* | STATS | SET03040 |
| C* | | SET03050 |
| | IF (J.NE.SRCD) GO TO 242 | SET03060 |
| | J=NXTCHR(CARD,COL) | SET03070 |
| | IF (J.EQ.TRCD) STATKY=1 | SET03080 |
| 242 | CONTINUE | SET03090 |
| C | CLUSTERS FOR MAPTAP | SET03100 |
| | IF (J.EQ.CPCD) MAPKEY=2 | SET03110 |
| C* | | SET03120 |
| C* | FIND12 A COMMA | SET03130 |
| C* | | SET03140 |
| 245 | J=FIND12(CARD,COL,COMVEC) | SET03150 |
| | IF (J.LE.0) GO TO 10 | SET03160 |

FILE: SETUP7

```

C      GO TO 240
C      SEQUENCE CARD
246     I=1
247     M=NXTCHR(CARD,COL)
        IF (M.EQ.BLANK) GO TO 248
        SEQUEN(I)=M
        I=I+1
        GO TO 247
248     NOSEQ = I - 1
        GO TO 10
C      PERCENT CARD
249     J=NXTCHR(CARD,COL)
        IF (J.EQ.BLANK) GO TO 10
        COL=COL+1
        J=NUMBER(CARD,COL,IPCT,J)
        PERCEN=1.-FLOAT(IPCT)/100.
        GO TO 10
C      MEANS CARD
250     J = NXTCHR(CARD,COL)
        IF (J .EQ. BLANK) GO TO 10
        IF (J .EQ. CRCD) GO TO 255
        IF (J .NE. FB CD) GO TO 10
        ISOKEY=1
        GO TO 10
255     ISOKEY=1
        CALL RDMEAN(ARRAY)
        GO TO 10
C*      READ MODULE DECK AND WRITE TO INPUT STAT UNIT AND FILE.
C*
C*      256     SAVE1=SAVTAP
                SAVE2=STAFIL
                SAVTAP=INUNIT
                STAFIL=INFILE
                CALL CRDSTA(ARRAY, TOP)
                SAVTAP=SAVE1
                STAFIL=SAVE2
                INITM=1
                GO TO 10
C      DATA FILE CARD
260     M = NXTCHR(CARD,COL)
        IF (M.EQ.BLANK) GO TO 10
        IF (M.EQ.URCD) GO TO 265
        IF (M.EQ.FRCD) GO TO 267
263     WRITE(6,750)
750     FORMAT(' ERROR ON DATA FILE CARD')
        GO TO 10
265     J = FIND12(CARD,COL,EQUVEC)
        IF (J.EQ.-1) GO TO 263
        M = NUMBER(CARD,COL,DATAPZ,ZERO)
        COL = COL - 1
        GO TO 260
267     J = FIND12(CARD,COL,EQUVEC)
        IF (J.EQ.-1) GO TO 263
        M = NUMBER(CARD,COL,DATFIL,ZERO)
        DATFIL = DATFIL - 1
        IF (DATFIL.LT. 0) DATFIL = 0
        COL = COL - 1
        GO TO 260
C      STAT FILE CARD
270     M=NXTCHR(CARD,COL)
271     IF (M.EQ.IP CD) GO TO 278
        IF (M.EQ.URCD) GO TO 275
        IF (M.EQ.FRCD) GO TO 277
        IF (M.EQ.ORCD) GO TO 272
        IF (M.EQ.COMMA) GO TO 270
        IF (M.EQ.BLANK) GO TO 10
273     WRITE(6,755)
755     FORMAT(' ERROR ON STATFILE CARD')

```

SET03170
 SET03180
 SET03190
 SET03200
 SET03210
 SET03220
 SET03230
 SET03240
 SET03250
 SET03260
 SET03270
 SET03280
 SET03290
 SET03300
 SET03310
 SET03320
 SET03330
 SET03340
 SET03350
 SET03360
 SET03370
 SET03380
 SET03390
 SET03400
 SET03410
 SET03420
 SET03430
 SET03440
 SET03450
 SET03460
 SET03470
 SET03480
 SET03490
 SET03500
 SET03510
 SET03520
 SET03530
 SET03540
 SET03550
 SET03560
 SET03570
 SET03580
 SET03590
 SET03600
 SET03610
 SET03620
 SET03630
 SET03640
 SET03650
 SET03660
 SET03670
 SET03680
 SET03690
 SET03700
 SET03710
 SET03720
 SET03730
 SET03740
 SET03750
 SET03760
 SET03770
 SET03780
 SET03790
 SET03800
 SET03810
 SET03820
 SET03830
 SET03840
 SET03850
 SET03860
 SET03870
 SET03880
 SET03890
 SET03900
 SET03910
 SET03920
 SET03930
 SET03940
 SET03950

FILE: SETUP7

| | | |
|-----|--|----------|
| | GO TO 10 | SET03960 |
| 272 | J=FINDI2(CARD,COL,SLASH) | SET03970 |
| | IF(J.EQ.-1)GO TO 273 | SET03980 |
| 274 | M=NXTCHR(CARD,COL) | SET03990 |
| | IF(M.EQ.COMMA) GO TO 274 | SET04000 |
| | IF(M.EQ.FHCD) GO TO 277 | SET04010 |
| | IF(M.NE.URCD) GO TO 271 | SET04020 |
| 275 | J=FINDI2(CARD,COL,EQUEVEC) | SET04030 |
| | IF(J.EQ.-1)GO TO 273 | SET04040 |
| | M=NUMBER(CARD,COL,SAVTAP,ZERO) | SET04050 |
| | COL=COL-1 | SET04060 |
| | GO TO 274 | SET04070 |
| 277 | J=FINDI2(CARD,COL,EQUEVEC) | SET04080 |
| | IF(J.EQ.-1)GO TO 273 | SET04090 |
| | M=NUMBER(CARD,COL,STAFIL,ZERO) | SET04100 |
| | COL=COL-1 | SET04110 |
| | STAFIL=STAFIL-1 | SET04120 |
| | IF(STAFIL.LT.0)STAFIL=0 | SET04130 |
| | GO TO 274 | SET04140 |
| 278 | J=FINDI2(CARD,COL,SLASH) | SET04150 |
| | IF(J.EQ.-1)GO TO 273 | SET04160 |
| | INITM = 1 | SET04170 |
| 279 | M=NXTCHR(CARD,COL) | SET04180 |
| | IF(M.EQ.COMMA) GO TO 279 | SET04190 |
| | IF(M.EQ.FHCD) GO TO 282 | SET04200 |
| | IF(M.NE.URCD) GO TO 271 | SET04210 |
| 281 | J=FINDI2(CARD,COL,EQUEVEC) | SET04220 |
| | IF(J.EQ.-1)GO TO 273 | SET04230 |
| | M=NUMBER(CARD,COL,INUNIT,ZERO) | SET04240 |
| | COL=COL-1 | SET04250 |
| | GO TO 279 | SET04260 |
| 282 | J=FINDI2(CARD,COL,EQUEVEC) | SET04270 |
| | IF(J.EQ.-1)GO TO 273 | SET04280 |
| | M=NUMBER(CARD,COL,INFILE,ZERO) | SET04290 |
| | COL=COL-1 | SET04300 |
| | GO TO 279 | SET04310 |
| C* | | SET04320 |
| C* | SUBCLASSES CARD--USE THE MEANS FOR THESE SUBCLASSES FROM THE | SET04330 |
| C* | STAT FILE FOR INITIAL MEANS | SET04340 |
| C* | | SET04350 |
| 283 | NOSUB2=NUMBER(CARD,COL,SUBVEC,NOSUB2) | SET04360 |
| | GO TO 10 | SET04370 |
| C* | | SET04380 |
| C* | MINIMUM POPULATION FOR STATISTICS PASS. | SET04390 |
| C* | | SET04400 |
| 284 | J=NXTCHR(CARD,COL) | SET04410 |
| | IF(J.NE.MINUS)COL=COL-1 | SET04420 |
| | M=NUMBER(CARD,COL,PMIN,ZERO) | SET04430 |
| | IF(J.EQ.MINUS)PMIN=0-PMIN | SET04440 |
| | GO TO 10 | SET04450 |
| C | DOTFIL INPUT/UNIT=N,FILE=M | SET04460 |
| C | OR UNIT=N,FILE=M | SET04470 |
| 300 | J=NXTCHR(CARD,COL) | SET04480 |
| 301 | IF(J.EQ.BLANK) GO TO 320 | SET04490 |
| | IF(J.NE.URCD) GO TO 305 | SET04500 |
| | J=FINDI2(CARD,COL,EQUEVEC) | SET04510 |
| | IF(J.NE.?) GO TO 320 | SET04520 |
| | ISTART=0 | SET04530 |
| | J=NUMBER(CARD,COL,ARRAY(TOP-30),ISTART) | SET04540 |
| | DOTUNT = ARRAY(TOP - 30) | SET04550 |
| | J=FINDI2(CARD,COL,EQUEVEC) | SET04560 |
| | IF(J.NE.?) GO TO 320 | SET04570 |
| | ISTART=0 | SET04580 |
| | J=NUMBER(CARD,COL,ARRAY(TOP-30),ISTART) | SET04590 |
| | DOTFIL=ARRAY(TOP-30) | SET04600 |
| | DOTFIL = DOTFIL - 1 | SET04610 |
| | GO TO 10 | SET04620 |
| 305 | IF(J.NE.IPCD) GO TO 310 | SET04630 |
| | J=FINDI2(CARD,COL,SLASH) | SET04640 |
| | IF(J.NE.?) GO TO 320 | SET04650 |
| | J = NXTCHR(CARD,COL) | SET04660 |
| | IF(J.EQ.FHCD) GO TO 315 | SET04670 |
| | IF(J.EQ.URCD) GO TO 301 | SET04680 |
| | GO TO 320 | SET04690 |
| 310 | IF(J.NE.FHCD) GO TO 320 | SET04700 |
| 315 | J = FINDI2(CARD,COL,EQUEVEC) | SET04710 |
| | IF(J.NE.?) GO TO 320 | SET04720 |
| | ISTART = 0 | SET04730 |
| | J = NUMBER(CARD,COL,ARRAY(TOP - 30),ISTART) | SET04740 |

FILE: SETUP7

```

DOTFIL = ARRAY(TOP - 30) - 1
J = FIND12(CARD,COL,EQUVEC)
IF (J.NE.2) GO TO 320
ISTART = 0
J = NUMBER(CARD,COL,ARRAY(TOP - 30),ISTART)
DOTUNT = ARRAY(TOP - 30)
GO TO 10
320 WRITE(6,760)
760 FORMAT(' ERROR ON DOTFILE CARD')
GO TO 10
C
340 SUNANG
J=NATCHR(CARD,COL)
IF (J.NE.TACD) GO TO 345
ISUNT=1
GO TO 10
345 ISTART=0
COL = COL - 1
J=NUMBER(CARD,COL,ARRAY(TOP-30),ISTART)
IF (J.GT.8) J = 8
IF (J.EQ.0) GO TO 10
ISUNC=0
DO 350 JJ=1,J
ISUNC=ISUNC+1
350 SUNANG(JJ)=ARRAY(TOP-31+ISUNC)
GO TO 10
C
380 DOTS
ISTART=0
J = NUMBER(CARD,COL,STDOTS(NSDOTS + 1),ISTART)
NSDOTS = NSDOTS + J
IF (NSDOTS.GT.60) NSDOTS=60
GO TO 10
C
C
*END* CARD
280 CONTINUE
IF (NOFEAT .GT. 0) GO TO 285
NOFEAT=30
DO 261 I=1,30
261 FETVEC(I)=I
285 CONTINUE
C*
C* PRINT USER REQUEST
C*
WRITE(6,660)
WRITE(6,670) ISTOP,NMIN,KRN,MAP,MAXCLS,NCLASS,
* (FETVEC(I),I=1,NOFEAT)
WRITE(6,680) DLMIN,STDMAX
WRITE(6,685) IPCT,PMIN,NSDOTS,ISUNC,ISUNT
IF (SPTRIG.EQ.1) WRITE(6,690) SEP
IF (ICHN.EQ.1) WRITE(6,710) CHNTHS
IF (PUNCH.EQ.1) WRITE(6,700)
IF (IORDER.EQ.1) WRITE(6,715)
IF (MAPFMT.EQ.1) WRITE(6,720)
IF (MAPFMT.EQ.2) WRITE(6,725)
IF (NOFEAT .GT. NMIN) WRITE(6,740)
RETURN
C
480 FORMAT(A4,6X,62A1)
490 FORMAT(' INVALID INPUT CARD--IGNORED'/T5,A4,6X,62A1)
500 FORMAT(10X,15A4)
510 FORMAT(10X,3A4)
550 FORMAT(7X,A4,4X,62A1)
630 FORMAT('/// INPUT SUMMARY'///)
640 FORMAT(' CHANNELS CANNOT BE CHANGED UNTIL THIS EXECUTION OF ISOCLS
* IS COMPLETED')
650 FORMAT(' NO. OF CLASSES CANNOT BE CHANGED UNTIL THIS EXECUTION OF
* ISOCLS IS COMPLETED')
660 FORMAT('/// YOU HAVE SELECTED THE FOLLOWING PARAMETER VALUES AND
* OPTIONS'///)
670 FORMAT(' STOP AFTER',I5,' ITERATION(S)' /
* ' ALLOW A MINIMUM OF',I6,' PIXELS PER CLUSTER' /
* ' PRINT A CLUSTER SUMMARY EVERY',I5,' ITERATION(S)' /
* ' PRINT A CLUSTER MAP EVERY',I5,' ITERATION(S)' /
* ' ALLOW A MAXIMUM OF',I5,' CLUSTERS PER CLASS' /
* ' THE STATISTICS FILE WILL BE WRITTEN AFTER',I4,' CLASS(ES)
* HAVE BEEN CLUSTERED' /
* ' CHANNELS ARE---',I3,30I3)
680 FORMAT(' DLMIN =',F7,3 / ' STDMAX=',F7,3)
685 FORMAT(' PERCENT =',I5 / ' IPCT =',I5 / ' NSDOTS =',I5 /

```

SET04750
SET04760
SET04770
SET04780
SET04790
SET04800
SET04810
SET04820
SET04830
SET04840
SET04850
SET04860
SET04870
SET04880
SET04890
SET04900
SET04910
SET04920
SET04930
SET04940
SET04950
SET04960
SET04970
SET04980
SET04990
SET05000
SET05010
SET05020
SET05030
SET05040
SET05050
SET05060
SET05070
SET05080
SET05090
SET05100
SET05110
SET05120
SET05130
SET05140
SET05150
SET05160
SET05170
SET05180
SET05190
SET05200
SET05210
SET05220
SET05230
SET05240
SET05250
SET05260
SET05270
SET05280
SET05290
SET05300
SET05310
SET05320
SET05330
SET05340
SET05350
SET05360
SET05370
SET05380
SET05390
SET05400
SET05410
SET05420
SET05430
SET05440
SET05450
SET05460
SET05470
SET05480
SET05490
SET05500
SET05510
SET05520
SET05530

FILE: SETUP7

```

      * NO. SUN ANGLES FROM CARDS = .15 / SUN ANGLE TAPE SW = .15/
690  FORMAT(1, SEP = . F7.3)
700  FORMAT(1, PUNCH THE MODULE STAT DECK)
710  FORMAT(1, CHAIN CLUSTERS WHICH ARE F7.3, UNITS APART)
715  FORMAT(1X, ORDER COLOR KEYS)
720  FORMAT(1X, WRITE A CLUSTER MAP OUTPUT TAPE IN UNIVERSAL FORMAT)
725  FORMAT(1X, WRITE A CLUSTER MAP OUTPUT TAPE IN LARSYS II FORMAT)
740  FORMAT(//, **WARNING** NMIN IS LESS THAN NO. OF CHANNELS, COVARIAN
      * CFS WILL NOT BE INVERTIBLE)
      END

```

SET05540
SET05550
SET05560
SET05570
SET05580
SET05590
SET05600
SET05610
SET05620
SET05630

FILE: SUNFAC

```

SUBROUTINE SUNFAC(SUNCOR,SUNANG,FETVEC,NOFEAT,ISUNC,ISUNT)
  INTEGER SUNANG,FETVEC,SUNA
  EQUIVALENCE (EXTRA(1),DUM1(1)),(EXTRA(109),DUM2(1)),
  * (EXTRA(217),DUM3(1))
  DIMENSION EXTRA(324),SUNANG(1),FETVEC(1),SUNCOR(1),
  * DUM1(108),DUM2(108),DUM3(108)
C
  INCLUDE COMMK6
  COMMON/GLOBAL/HEAD(63),MAPTAP,DATEPE,SAVTAP,RMFILE,RMKEY,
  * HISFIL,HISKEY,TRFORM,EHPTP,EPPKEY,MAPUNT,NOFILE,
  * DRUMAD,DRM-OS,PAGSIZ,DATEFIL,STAFIL,ASAV,ASAVL
  * ,NHSTUN,NHSTFI,SCTRUN,MAPFIL
  * ,DOTUNT,DOTFIL,NCHPAS,TRNSFL,BMTRFL,HISTFL,PCHUNT,
  * CRDUNT,PRUNT,RANDIO
CSEND
C
  DATA DUM1 /16.413,14.887,14.089,13.401,
  * 13.015,11.901,11.323,10.816,10.755,9.893,9.449,9.052,
  * 9.094,8.413,8.063,7.745,7.832,7.285,7.002,6.744,
  * 6.880,6.425,6.140,5.973,6.098,5.722,5.527,5.346,
  * 5.474,5.156,4.991,4.838,4.967,4.694,4.551,4.419,
  * 4.530,4.297,4.174,4.061,
  * 4.163,3.961,3.855,3.757,3.852,3.675,3.583,3.497,
  * 3.580,3.426,3.344,3.269,3.342,3.207,3.135,3.069,
  * 3.135,3.015,2.952,2.893,2.953,2.846,2.789,2.737,
  * 2.788,2.693,2.643,2.597,2.642,2.557,2.513,2.471,
  * 2.511,2.435,2.395,2.358,
  * 2.393,2.289,2.255,2.285,2.223,2.191,2.161,
  * 2.187,2.131,2.102,2.076,2.097,2.047,2.021,1.997,
  * 2.016,1.971,1.947,1.925,1.940,1.899,1.878,1.858,
  * 1.871,1.834,1.815,1.797,1.807,1.773,1.756,1.740/
  DATA DUM2 /1.748,1.717,1.701,1.687,
  * 1.693,1.665,1.651,1.637,1.642,1.617,1.603,1.591,
  * 1.594,1.571,1.559,1.548,1.550,1.529,1.518,1.508,
  * 1.508,1.489,1.479,1.470,1.469,1.452,1.443,1.435,
  * 1.432,1.417,1.409,1.402,1.398,1.384,1.377,1.370,
  * 1.366,1.353,1.347,1.341,
  * 1.336,1.324,1.318,1.313,1.307,1.297,1.291,1.286,
  * 1.280,1.271,1.266,1.261,1.255,1.246,1.242,1.238,
  * 1.230,1.223,1.219,1.216,1.208,1.201,1.198,1.195,
  * 1.186,1.180,1.177,1.174,1.166,1.161,1.158,1.155,
  * 1.146,1.142,1.140,1.137,
  * 1.128,1.124,1.122,1.120,1.111,1.107,1.106,1.104,
  * 1.094,1.091,1.090,1.089,1.078,1.076,1.075,1.074,
  * 1.064,1.062,1.061,1.060,1.050,1.048,1.047,1.047,
  * 1.036,1.035,1.035,1.034,1.023,1.023,1.022,1.022/
  DATA DUM3 /1.011,1.011,1.011,1.011,
  * 1.000,1.000,1.000,1.000,.989,.989,.990,.990,
  * .979,.979,.980,.980,.969,.970,.971,.971,
  * .960,.961,.961,.962,.951,.952,.953,.954,
  * .943,.944,.945,.946,.935,.937,.937,.938,
  * .929,.929,.930,.931,
  * .921,.923,.924,.925,.914,.916,.917,.918,
  * .908,.910,.911,.912,.902,.904,.906,.907,
  * .897,.899,.901,.902,.892,.894,.896,.897,
  * .887,.890,.891,.892,.882,.885,.887,.888,
  * .878,.881,.883,.884,
  * .875,.878,.879,.881,.871,.874,.876,.878,
  * .868,.871,.873,.875,.865,.869,.870,.872,
  * .863,.866,.868,.870,.861,.864,.866,.868,
  * .850,.862,.864,.866,.857,.861,.863,.864/
  DATA PLANK /1/
  KS = 0
  DO 200 I=1,NOFEAT
    K=(FETVEC(I)-1)/NCHPAS
    IF (ISUNC.NE.0.AND.I.EQ.1) KS=K
    KR=FETVEC(I)-K*NCHPAS
    K=K+1
    SUNA = SUNANG(K-KS)
  100  JND=(SUNA-5)*NCHPAS+KR
    SUNCOR(I)=EXTRA(JND)
  200  CONTINUE
    WRITE(6,90)
  90  FORMAT(/T61,'SUN ANGLES:')
    WRITE(6,210) (SUNANG(I), I = 1,8)
  210  FORMAT(T45,A15)
    WRITE(6,215)
  215  FORMAT(/T52,'CORRECTIONS FOR SUN ANGLES:')
C
  NOFETR = NOFEAT

```

SUN00010
 SUN00020
 SUN00030
 SUN00040
 SUN00050
 SUN00060
 SUN00070
 SUN00080
 SUN00090
 SUN00100
 SUN00110
 SUN00120
 SUN00130
 SUN00140
 SUN00150
 SUN00160
 SUN00170
 SUN00180
 SUN00190
 SUN00200
 SUN00210
 SUN00220
 SUN00230
 SUN00240
 SUN00250
 SUN00260
 SUN00270
 SUN00280
 SUN00290
 SUN00300
 SUN00310
 SUN00320
 SUN00330
 SUN00340
 SUN00350
 SUN00360
 SUN00370
 SUN00380
 SUN00390
 SUN00400
 SUN00410
 SUN00420
 SUN00430
 SUN00440
 SUN00450
 SUN00460
 SUN00470
 SUN00480
 SUN00490
 SUN00500
 SUN00510
 SUN00520
 SUN00530
 SUN00540
 SUN00550
 SUN00560
 SUN00570
 SUN00580
 SUN00590
 SUN00600
 SUN00610
 SUN00620
 SUN00630
 SUN00640
 SUN00650
 SUN00660
 SUN00670
 SUN00680
 SUN00690
 SUN00700
 SUN00710
 SUN00720
 SUN00730
 SUN00740
 SUN00750
 SUN00760
 SUN00770
 SUN00780
 SUN00790

FILE: SUNFAC

```
      ISTART = 1
      IEND = 16
217  IF (IEND .GE. NOFETR) IEND = NOFETR
      IENDS = ISTART + IEND - 1
      WRITE(6,220) (HLANK,FETVEC(I),I=ISTART,IENDS)
220  FORMAT(//1X,16(A1,'CH(':12.1)'):1X))
      WRITE(6,230) (SUNCOR(I),I=ISTART,IENDS)
230  FORMAT( 16(2X,F6.4))
C
      NOFETR = NOFFTR - IEND
      ISTART = IENDS + 1
      IF (NOFETR .LE. 0) RETURN
      GO TO 217
      END
```

SUN00800
SUN00810
SUN00820
SUN00830
SUN00840
SUN00850
SUN00860
SUN00870
SUN00880
SUN00890
SUN00900
SUN00910
SUN00920
SUN00930

FILE TAPHDR

```

SUBROUTINE TAPHDR(DATAPE,IFILE)
IMPLICIT INTEGER(A-Z)
.....
TAPERD READS THE MULTISPECTRAL SCANNER DATA TAPE, UNPACKS THE
REQUESTED DATA AND RETURNS IT UNPACKED TO THE CALLING ROUTINE.
THERE ARE THREE SEPARATE SUBROUTINES: TAPHDR, FLDINT
AND LINERD, NEEDED TO READ A TAPE
TAPHDR MUST BE CALLED ONCE TO READ THE HEADER RECORD AND UNPACK
NECESSARY DATA FROM THE RECORD
CALL TAPHDR(DATAPE,IFILE)
DATAPE=INPUT UNIT NUMBER FOR DATA TAPE
IFILE = NO. OF E-O-F'S ON DATA TAPE USER WISHES TO READ OVER
IN ORDER TO POSITION TAPE TO DESIRED FILE
FLDINT MUST BE CALLED ONCE FOR EACH FIELD. THE TAPE IS POSITIONED
TO THE CORRECT RECORD AND PARAMETERS ARE INITIALIZED FOR THE FIELD
CALL FLDINT(BLOCK,FETVEC,NOFEAT)
BLOCK(1)=LINE START
BLOCK(2)=LINE END
BLOCK(3)=LINE INCREMENT
BLOCK(4)=SAMPLE START
BLOCK(5)=SAMPLE END
CONTINUE
BLOCK(6)=SAMPLE INCREMENT
FETVEC= (INPUT) VECTOR CONTAINING FEATURES REQUESTED
NOFEAT (INPUT) NO. OF FEATURES IN FETVEC
CALL LINERD(IDATA,ENDTAP)
IDATA=(OUTPUT) ARRAY CONTAINING UNPACKED DATA
ENDTAP = TRIGGER INDICATING WHETHER OR NOT AN E-O-F HAS BEEN
REACHED WHILE TRYING TO READ A GIVEN SCAN LINE NO.
IF AN E-O-F IS FOUND ENDTAP = -1, OTHERWISE, ENDTAP
= 0.
.....
READY IS A INDICATOR TO TEST WHETHER THE TAPE HAS BEEN
POSITIONED AND PARAMETERS SET FOR A FIELD
THE APRAYS NR AND HWRD ARE PRECALCULATED WORD AND BIT
POSITIONS OF INFORMATION IN THE HEADER RECORD OF THE UNIVERSAL
FORMAT WHICH MUST BE EXTRACTED.
CONTINUE
NRPDS = NO. OF RECORDS PER DATA SET
NCPW = NO. OF CHANNELS PER RECORD ON RECORDS PAST ANCILLARY RECORD
NPRC = NO. OF PHYSICAL RECORDS PER CHANNEL
ANCLNG = ANCILLARY LENGTH IN BYTES
NC = NO. OF CHANNELS
NS = NO. OF SAMPLES PER CHANNEL PER SCAN
NHITS = NO. OF BITS PER PIXEL
DOI = DATA ORDER INDICATOR
NDSPH = NO. OF DATA SETS PER RECORD
NCAR = NO. OF CHANNELS OF VIDEO DATA ON SAME RECORD
WITH ANCILLARY DATA)
SVD = START OF VIDEO DATA. (BYTE POSITION WITHIN DATA FOR
A GIVEN CHANNEL)
LOGICAL*1 IINH(30600)
LOGICAL*1 IROWD(4), ILINE(4)
DIMENSION FNM(3,4)
DIMENSION N4(24), HWRD(24)
COMMON /TAPHDR/ IUNIT,IFIRST,FSCAN,SAMEND,SAMINC,READY,NSCAN,
* LINC,ID(200),ISL,ILHUF(30),JREC(30),IHYTE(30),NBUFS,FILENO,LINEND
* LININC,NSAMP,NCHAN,FORMAT
COMMON /IDSTOR/IDD(250)
COMMON /ISOLK/SUNANG(8),ISUNT,ISUNC,SMSTR,SMSTP,SMINC,LINSKP
EQUIVALENCE (IROWD,40WD)
EQUIVALENCE (IFIRST,ILINE)
EQUIVALENCE (ID(1),NRPDS ),(ID(2),NCPW ),
* (ID(3),NPRC ),(ID(4),ANCLNG),
* (ID(5),NC ),(ID(6),NS )

```

TAP00010
TAP00020
TAP00030
TAP00040
TAP00050
TAP00060
TAP00070
TAP00080
TAP00090
TAP00100
TAP00110
TAP00120
TAP00130
TAP00140
TAP00150
TAP00160
TAP00170
TAP00180
TAP00190
TAP00200
TAP00210
TAP00220
TAP00230
TAP00240
TAP00250
TAP00260
TAP00270
TAP00280
TAP00290
TAP00300
TAP00310
TAP00320
TAP00330
TAP00340
TAP00350
TAP00360
TAP00370
TAP00380
TAP00390
TAP00400
TAP00410
TAP00420
TAP00430
TAP00440
TAP00450
TAP00460
TAP00470
TAP00480
TAP00490
TAP00500
TAP00510
TAP00520
TAP00530
TAP00540
TAP00550
TAP00560
TAP00570
TAP00580
TAP00590
TAP00600
TAP00610
TAP00620
TAP00630
TAP00640
TAP00650
TAP00660
TAP00670
TAP00680
TAP00690
TAP00700
TAP00710
TAP00720
TAP00730
TAP00740
TAP00750
TAP00760

FILE TAPHDR

```

*          (ID(7),NMITS),(ID(8),DOI),
*          (ID(9),NDSPH),(ID(10),NCAR),
*          (ID(11),SVD),(ID(12),PRSZ)
* DATA FRM/UNIV,FRSA,IL,LARS,YS 2,
*          LAND,SAT,1/2,LAND,SAT,1/3,
* DATA HWRD/104.102.103.105.90.1747.91.107.1774.1785.92.104.110.
*          1789.1791.100.2201.2203.2205.2207.2209.2211.2213.2215.
* 61.62.63.67/
* DATA NR/8.A.16.A.16.3.8.A.16.16.16.16.16.16.
* 16.16.16.16.16.16.16.16.8.8.8.16/

ENTRY FOR READING HEADER INFORMATION

INFORMATION IN ERCDIC OR IIR FLOAING POINT IS NOT UNPACKED
FROM THE HEADER RECORD AT THIS TIME.

READY = -1
IFIRST = 0
FILENO = IFILE
IUNIT=DATAPE
REWIND IUNIT

SKIP DIRECTORY FILE FOR LANDSAT 3

IF (FORMAT.NE.4) GO TO 2
HEAD(IUNIT,510,END=4) DUMMY
GO TO 3
FILENO=FILENO*3
REC = 0
KHUF = 3060
IF (ISUNC.NE.0) GO TO 6
DO 5 I = 1,9
SUMANG(I) = 60
CONTINUE

NRPOS=1
IF (FILENO.EQ.0) GO TO 600
DO A I=1,FILENO
IF (FORMAT.NE.4) WRITE(6,610) I
610 FORMAT(' SKIPPING FILE',I8)
IF (FORMAT.NE.4) GO TO 7
IJ=I/3
IF (MOD(I,3).EQ.0) WRITE(6,610) IJ
7 HEAD(IUNIT,510,END=8) DUMMY
GO TO 7
CONTINUE
600 IF (FORMAT.EQ.2) GO TO 11
IF (FORMAT.EQ.3) GO TO 1000
IF (FORMAT.EQ.4) GO TO 1000
9 CALL HUFILL(WEC,IUNIT,KHUF,IHUF,NRPOS,ENDTAP,IERR)
IF (IERR.EQ.-1) GO TO 11
FORMAT = 1
GO TO 20
11 KHUF = 800
REC = 0
CALL HUFILL(WEC,IUNIT,KHUF,1D,NRPOS,ENDTAP,IERR)
IF (IERR.EQ.-1) GO TO 10
FORMAT = 2
GO TO 20
10 WRITE(6,280)
WRITE(6,340)
CALL CMERR
20 IF (FORMAT.EQ.1) GO TO 40
IF (FORMAT.NE.2) GO TO 265
DO 12 I=4,200
12 ID(I)=ID(1)
NCS=NC*NS
PRSZ = NCS * 4
IF (PRSZ.LE.30600) GO TO 35
WRITE(6,330) NC*NS
CALL CMERR
35 CONTINUE
NRPOS=1
MAXREC = PRSZ

```

TAP00770
TAP00780
TAP00790
TAP00800
TAP00810
TAP00820
TAP00830
TAP00840
TAP00850
TAP00860
TAP00870
TAP00880
TAP00890
TAP00900
TAP00910
TAP00920
TAP00930
TAP00940
TAP00950
TAP00960
TAP00970
TAP00980
TAP00990
TAP01000
TAP01010
TAP01020
TAP01030
TAP01040
TAP01050
TAP01060
TAP01070
TAP01080
TAP01090
TAP01100
TAP01110
TAP01120
TAP01130
TAP01140
TAP01150
TAP01160
TAP01170
TAP01180
TAP01190
TAP01200
TAP01210
TAP01220
TAP01230
TAP01240
TAP01250
TAP01260
TAP01270
TAP01280
TAP01290
TAP01300
TAP01310
TAP01320
TAP01330
TAP01340
TAP01350
TAP01360
TAP01370
TAP01380
TAP01390
TAP01400
TAP01410
TAP01420
TAP01430
TAP01440
TAP01450
TAP01460
TAP01470
TAP01480
TAP01490
TAP01500
TAP01510
TAP01520

FILE TAPHDR

```

NCAR=NC
ANCLNG=4
SVD=1
NBITS=8
DOI=0
NCPR=0
NDSPR=1
NPRC = 0
WRITE(6,481) (FRM(I,2),I=1,3),NC,NS
SMSTR=1
GO TO 100

C*
C* UNPACK NECESSARY INFORMATION FROM HEADER RECORD-UNIVERSAL FORMAT
C*
40  ILIM = 156
    DO 60 I = 1,ILIM
      IWD = 112 * (I - 29) * 4
      IF (I.LT.29) IWD = MWORD(I)
      WORD = 0
      NBYTES = 4
      IF (I.LT.29) NBYTES = NB(I)/8
      DO 55 J=1,NBYTES
        LOC = 4 * J - NBYTES
        IPDS=IWD+J-1
55    IWORD(LOC) = IBUF(IWD+J-1)
        ID(I) = WORD
60    CONTINUE
      SMSTR = ID(12)
      SMSTP = ID(13)
      SMINC = ID(14)
      LINSKP = ID(15)
      IF (ISUNT.EQ.0) GO TO 65
      DO 62 I=1,8
        IF (ID(16+I).LT.5.OR.ID(16+I).GT.85) ID(16+I) = 60
62    SUNANG(I) = ID(16+I)
65    CONTINUE
      WORD = 0
      DO 66 I=1,4
        IPAT = 2254 * (I - 1) * 8
        IWORD(4) = IBUF(IPAT)
        ID(159 + I) = WORD
66
C***
C*** CODE JUST ABOVE ADDED OCT.20,1978 TO UNPACK SOIL LINES
C***
      IF (ISUNT.GT.0) WRITE(6,482) (SUNANG(I), I=1,8)
482 FORMAT(14// 'SUN ANGLES : ',8I6)
      WRITE(6,481) (FRM(I,1),I=1,3),NC,NS
      MAXREC = PRG2
70  IF (NPRC.LF.1) GO TO 80
      WRITE(6,360)
      CALL CMERR
80  CONTINUE
      IF (SVD.LE.0) SVD=1
      IF (NDSPR.LE.0) NDSPR=1
      IF (NBITS.EQ.8) GO TO 90
      WRITE(6,390) NBITS
      NBITS=8
90  IF (DOI.EQ.0) GO TO 100
      WRITE(6,400) DOI
      CALL CMERR
100 CONTINUE
      KPTS=0
      IPD=0

C*
C* DATA SET LENGTH IN BYTES
C* DSL=ANCLNG*NS*NC
C*
C* READ FIRST DATA SET TO DETERMINE FIRST SCAN LINE NUMBER
C*
      IBUF=1
      RECF=0
      CALL BUFILL(PEC,IUNIT,MAXREC,IBUF,NRPDS,ENDTAP,IERR)
C
C IFIRST = ILINE(1-4)
C
      IF (FORMT.EQ.1) ILINE(3) = IBUF(71)

```

TAP01530
 TAP01540
 TAP01550
 TAP01560
 TAP01570
 TAP01580
 TAP01590
 TAP01600
 TAP01610
 TAP01620
 TAP01630
 TAP01640
 TAP01650
 TAP01660
 TAP01670
 TAP01680
 TAP01690
 TAP01700
 TAP01710
 TAP01720
 TAP01730
 TAP01740
 TAP01750
 TAP01760
 TAP01770
 TAP01780
 TAP01790
 TAP01800
 TAP01810
 TAP01820
 TAP01830
 TAP01840
 TAP01850
 TAP01860
 TAP01870
 TAP01880
 TAP01890
 TAP01900
 TAP01910
 TAP01920
 TAP01930
 TAP01940
 TAP01950
 TAP01960
 TAP01970
 TAP01980
 TAP01990
 TAP02000
 TAP02010
 TAP02020
 TAP02030
 TAP02040
 TAP02050
 TAP02060
 TAP02070
 TAP02080
 TAP02090
 TAP02100
 TAP02110
 TAP02120
 TAP02130
 TAP02140
 TAP02150
 TAP02160
 TAP02170
 TAP02180
 TAP02190
 TAP02200
 TAP02210
 TAP02220
 TAP02230
 TAP02240
 TAP02250
 TAP02260
 TAP02270
 TAP02280

FILE TAPHDR

| | | |
|------|---|----------|
| | IF (FORMT.EQ.1) ILINE(4) = IBUF(72) | TAP02290 |
| | IF (FORMT.EQ.2) ILINE(3) = IBUF(1) | TAP02300 |
| | IF (FORMT.EQ.2) ILINE(4) = IBUF(2) | TAP02310 |
| | IF (IFRST.GT.0) GO TO 120 | TAP02320 |
| | WRITE (6,300) | TAP02330 |
| | WRITE (6,340) | TAP02340 |
| | CALL CMERR | TAP02350 |
| 120 | FSCAN=IFRST | TAP02360 |
| | WRITE (6,500) IFRST, SMSTR | TAP02370 |
| | RETURN | TAP02380 |
| 265 | WRITE (6,340) | TAP02390 |
| | CALL CMERR | TAP02400 |
| | RETURN | TAP02410 |
| C | | TAP02420 |
| C | SET UP FOR LANDSAT 1 OR 2 FORMAT | TAP02430 |
| C | | TAP02440 |
| 1000 | KBUF=40 | TAP02450 |
| | REC=0 | TAP02460 |
| | CALL BUFill (REC, IUNIT, KBUF, IBUF, NRPDS, ENDTAP, IERR) | TAP02470 |
| | IF (IERR.EQ.-1) GO TO 10 | TAP02480 |
| C | | TAP02490 |
| C | UNPACK DATA FROM LANDSAT 1 OR 2 HEADER | TAP02500 |
| C | | TAP02510 |
| | NRITS=8 | TAP02520 |
| | DOI=2 | TAP02530 |
| | NRPDS=1 | TAP02540 |
| | NCPR=4 | TAP02550 |
| | NPRC=1 | TAP02560 |
| | ANCLNG=0 | TAP02570 |
| | NC=4 | TAP02580 |
| | WORD=0 | TAP02590 |
| | IWORD(3)=IBUF(39) | TAP02600 |
| | IWORD(4)=IBUF(40) | TAP02610 |
| | NS=WORD/4 | TAP02620 |
| | NDSPR=1 | TAP02630 |
| | NCAR=4 | TAP02640 |
| | SVD=1 | TAP02650 |
| | IWORD(3)=IBUF(17) | TAP02660 |
| | IWORD(4)=IBUF(18) | TAP02670 |
| | PKSZ=WORD | TAP02680 |
| | DSL=NS*NC | TAP02690 |
| | FSCAN=1 | TAP02700 |
| | IFRST=1 | TAP02710 |
| | SMSTR=1 | TAP02720 |
| | WRITE (6,481) (IRM(I,3), I=1, NC, NS | TAP02730 |
| | WRITE (6,500) IFRST, SMSTR | TAP02740 |
| | RETURN | TAP02750 |
| C | | TAP02760 |
| C | SET UP FOR LANDSAT 3 | TAP02770 |
| C | | TAP02780 |
| 2000 | KBUF=3596 | TAP02790 |
| | REC=0 | TAP02800 |
| | CALL BUFill (REC, IUNIT, KBUF, IBUF, NRPDS, ENDTAP, IERR) | TAP02810 |
| | IF (IERR.EQ.-1) GO TO 10 | TAP02820 |
| | NRITS=8 | TAP02830 |
| | DOI=0 | TAP02840 |
| | WORD=0 | TAP02850 |
| | IWORD(4)=IBUF(120) | TAP02860 |
| C | | TAP02870 |
| C | TYPE INDICATOR 0=SEQUENTIAL 1=INTERLEAVED | TAP02880 |
| C | | TAP02890 |
| | TYPE=0 | TAP02900 |
| | IF (WORD.NE.0) TYPE=1 | TAP02910 |
| | IF (TYPE.EQ.0) NRPDS=1 | TAP02920 |
| | NCPR=1 | TAP02930 |
| | NPRC=1 | TAP02940 |
| | ANCLNG=0 | TAP02950 |
| | IF (TYPE.EQ.0) NC=1 | TAP02960 |
| | IF (TYPE.EQ.0) GO TO 2200 | TAP02970 |
| C | | TAP02980 |
| C | SET NC AND NRPDS FOR INTERLEAVED FORMAT | TAP02990 |
| C | | TAP03000 |
| | WORD=0 | TAP03010 |
| | IWORD(4)=IBUF(46) | TAP03020 |
| | IF (WORD.EQ.3) NRPDS=5 | TAP03030 |
| | IF (WORD.NE.3) NRPDS=4 | TAP03040 |

FILE TAPHDR

| | | |
|------|---|----------|
| 2200 | NC=NRPDS | TAP03050 |
| | WORD=0 | TAP03060 |
| | IWORD(3)=IBUF(131) | TAP03070 |
| | IWORD(4)=IBUF(132) | TAP03080 |
| | NS=WORD | TAP03090 |
| | NDSR=1 | TAP03100 |
| | NCAR=1 | TAP03110 |
| | SVN=13 | TAP03120 |
| | PRSZ=3596 | TAP03130 |
| | DSI=3596 | TAP03140 |
| | JREC(1)=TYPE | TAP03150 |
| C | | TAP03160 |
| C | SKIP REMAINDER OF HEADER FILE | TAP03170 |
| 2210 | RFAD(IUNIT,510,END=2220)DUMMY | TAP03180 |
| | GO TO 2210 | TAP03190 |
| 2220 | CONTINUE | TAP03200 |
| C | | TAP03210 |
| | FSCAN=1 | TAP03220 |
| | IFRST=1 | TAP03230 |
| | SMSTR=1 | TAP03240 |
| | WRITE(6,481)(FRM(I,4),I=1,3),NC,NS | TAP03250 |
| | WRITE(6,500)IFRST,SMSTR | TAP03260 |
| | RETURN | TAP03270 |
| 240 | FORMAT(' UNRECOVERABLE ERROR READING HEADER RECORD') | TAP03280 |
| 300 | FORMAT(' A LINE NO. IS LESS THAN OR EQUAL ZERO') | TAP03290 |
| 310 | FORMAT(' LAST SCAN LINE READ',I5,' ISTAT=',I5) | TAP03300 |
| 330 | FORMAT(' INTERNAL DIMENSIONS TOO SMALL FOR DATA',I7,' NO. OF CHANNELS',I7) | TAP03310 |
| | *S ON DATA TAPE=',I7,' NO. OF POINTS/CHANNEL=',I7/) | TAP03320 |
| 340 | FORMAT(' CHECK THE FOLLOWING POSSIBLE ERRORS',I7,' 1. DATA TAPE IS NOT') | TAP03330 |
| | *T IN REQUESTED FORMAT') | TAP03340 |
| 360 | FORMAT(' ONLY ONE OR LESS RECORDS PER CHANNEL ACCEPTABLE AT THIS TAPE') | TAP03350 |
| | *IME') | TAP03360 |
| 370 | FORMAT(' NO. OF RECORDS PER DATA SET=',I5,' MUST BE LESS THAN OR EQUAL',I5) | TAP03370 |
| | *QUAL',I5/) | TAP03380 |
| 390 | FORMAT(' NO. OF BITS/PIXEL=',I5,' ONLY 8 BITS ACCEPTABLE AT THIS TAPE') | TAP03390 |
| | *IME') | TAP03400 |
| 400 | FORMAT(' DATA ORDER INDICATOR=',I5,' DATA MUST BE ORDERED BY PIXEL',I5) | TAP03410 |
| | *') | TAP03420 |
| 481 | FORMAT(1H '///' INPUT IMAGE DATA TAPE INFORMATION'// | TAP03430 |
| | * 5X,'FORMAT',T30,3A4/ | TAP03440 |
| | * 5X,'NO. OF CHANNELS',T30,I4 / | TAP03450 |
| | * 5X,'NO. OF PIXELS/LINE',T30,I4/ | TAP03460 |
| 500 | FORMAT(5X,'FIRST SCAN LINE NO.',T30,I4/ | TAP03470 |
| | * 5X,'FIRST PIXEL REFERENCE PT.',T30,I4/ | TAP03480 |
| 510 | FORMAT(1A4) | TAP03490 |
| | END | TAP03500 |
| | | TAP03510 |

ORIGINAL PAGE IS
OF POOR QUALITY

FILE: WRTBMT

```

SUBROUTINE WRTBMT(BMAT,NOFET4,NOFET2,FETVC2)
DIMENSION CH(2)
DATA CH/'CH( 1, 1 )' /
INTEGER FETVC2(30)
C INCLUDE COMPKA.LIST
COMMON/GLOBAL/HEAD(63),MAPTAP,DATAP,SAVTAP,RMFILE,RMKEY,
* HISFIL,HISKEY,TRFORM,ENIPTP,ERPKEY,MAPUNT,NOFILF,
* DRUMAD,DRUMDS,PAGSIZ,DATFIL,STAFIL,ASAV,ASAVFL
* ,NHSTUN,NHSTFI,SCTHUN,MAPEIL
* ,DOTUNT,DOTFIL,NCHPAS,TRNSFL,BMTRFL,HISTFL,PCHUNT,
* CRDUNT,PRJUNT,RANDIO
CSEND
DIMENSION BMAT(NOFET4,NOFET2)
DOUBLE PRECISION BMAT
GO TO 4
ENTRY WRTM(RBMAT,NOFET4,NOFET2,FETVC2)
DIMENSION RBMAT(NOFET4,NOFET2)
K=1
4 CONTINUE
WRITE(6,HEAD)
WRITE(6,100)NOFET4,NOFET2
IR=1
IK=12
5 IF(IK.GT.NOFET2)IK=NOFET2
WRITE(6,200)(CH(1),CH(2), I=IR,IK)
WRITE(6,300)(FETVC2(I),I=IR,IK)
WRITE(6,350)
IF(K.EQ.0) GO TO 11
DO 12 J=1,NOFET4
12 WRITE(6,400) J,(BMAT(J,I),I=IR,IK)
GO TO 13
11 CONTINUE
DO 10 J=1,NOFET4
10 WRITE(6,400) J,(BMAT(J,I),I=IR,IK)
13 CONTINUE
IF(IK.EQ.NOFET2)GO TO 20
IR=IK+1
IK=IK+12
GO TO 5
20 RETURN
100 FORMAT(///45X,'LINEAR TRANSFORMATION (B) MATRIX'//
* 50X,'NO. LINEAR COMB. -',I3/
* 50X,'NO. CHANNELS -',I3/)
200 FORMAT(/12X,12(A4,A4,2X))
300 FORMAT(1F+,14X,11(I2,8X),I2)
350 FORMAT(1X,'LIN. COMB.')
400 FORMAT(1X,I5,4X,12(1X,F9.3))
END

```

WRT00010
WRT00020
WRT00030
WRT00040
WRT00050
WRT00060
WRT00070
WRT00080
WRT00090
WRT00100
WRT00110
WRT00120
WRT00130
WRT00140
WRT00150
WRT00160
WRT00170
WRT00180
WRT00190
WRT00200
WRT00210
WRT00220
WRT00230
WRT00240
WRT00250
WRT00260
WRT00270
WRT00280
WRT00290
WRT00300
WRT00310
WRT00320
WRT00330
WRT00340
WRT00350
WRT00360
WRT00370
WRT00380
WRT00390
WRT00400
WRT00410
WRT00420
WRT00430
WRT00440
WRT00450
WRT00460
WRT00470
WRT00480

FILE: WRTDOT

| | | |
|---|--|----------|
| C | | WRT00010 |
| C | DOTFIL OUTPUTS THE DOT DATA FILE | WRT00020 |
| C | A FILE IS CREATED FOR TYPE OF DOTS | WRT00030 |
| C | SUBROUTINE WRTDOT(TOTDOT,NOSUN,FLDSAV,VERTEX,ANGLE,DOTS,NOCAT, | WRT00040 |
| C | CATNAM,SIZE,NOFET2,FETVC2,TOTVT2,NOFLD2, | WRT00050 |
| C | UNIT,FILE) | WRT00060 |
| C | | WRT00070 |
| C | IMPLICIT INTEGER (A-Z) | WRT00080 |
| C | DIMENSION CATNAM(NOCAT),FLDSAV(4,1),VERTEX(2,1),ANGLE(1) | WRT00090 |
| C | DIMENSION FETVC2(30),DOTS(SIZE,TOTDOT) | WRT00100 |
| C | DOTFIL = FILE | WRT00110 |
| C | DOTUNT = UNIT | WRT00120 |
| C | POSITION TO DESIRED FILE | WRT00130 |
| C | REWIND DOTUNT | WRT00140 |
| C | CALL FSBSFL(DOTUNT,DOTFIL,ISTAT) | WRT00150 |
| C | REC NO. 1 -- INDICES FOR REC NO. 2 | WRT00160 |
| C | WRITE(DOTUNT)NOCAT,NOFET2,NOFLD2,TOTVT2,TOTDOT,NOSUN,(CATNAM(I), | WRT00170 |
| C | I=1,NOCAT),SIZE | WRT00180 |
| C | REC NO. 2 | WRT00190 |
| C | WRITE(DOTUNT)(FETVC2(I),I=1,NOFET2),((FLDSAV(I,J),I=1,4),J=1, | WRT00200 |
| C | NOFLD2),((VERTEX(I,J),I=1,2),J=1,TOTVT2),(ANGLE(I),I=1,NOSUN) | WRT00210 |
| C | REC NO. 3 -- DOT DATA INFO | WRT00220 |
| C | WRITE(DOTUNT)((DOTS(I,J),I=1,SIZE),J=1,TOTDOT) | WRT00230 |
| C | END FILE DOTUNT | WRT00240 |
| C | RETURN | WRT00250 |
| C | END | WRT00260 |
| | | WRT00270 |
| | | WRT00280 |
| | | WRT00290 |
| | | WRT00300 |
| | | WRT00310 |
| | | WRT00320 |
| | | WRT00330 |
| | | WRT00340 |
| | | WRT00350 |
| | | WRT00360 |
| | | WRT00370 |
| | | WRT00380 |
| | | WRT00390 |

FILE: WRTFLD

```

SUBROUTINE WRTFLD(FLDSAV,VERTEX,NOFLD,KEY,CLSNAM,SUBNAM)
IMPLICIT INTEGER(A-Z)
DIMENSION CLSNAM(1),SUBNAM(1)
C*
C* THIS SUBROUTINE PRINTS SAVED TRAINING OR TEST FIELDS
C*
C* DIMENSION FLDSAV(4,NOFLD),VERTEX(2,1)
C* DATA LPRN/1(1)/
C INCLUDE COMRK6.LIST
COMMON/GLOBAL/HEAD(63),MAPTAP,DATAPE,SAVTAP,BMFILE,BMKEY,
* HISFIL,HISKEY,TRFORM,ERIPTP,ERPKEY,MAPUNT,NOFILE,
* DRUMAD,BRMWDS,PAGSIZ,DATFIL,STAFIL,ASAV,ASAVFL
* ,NHSTUN,NHSTFI,SCTHUN,MAPFIL
* ,DOTUNT,DOTFIL,NCHPAS,TRNSFL,BMTRFL,HISTFL,PCHUNT,
* CROUNT,PRUNT,RANDIO
C*END
IR=1
WRITE(6,HEAD)
IF(KEY.EQ.1)WRITE(6,100)
IF(KEY.EQ.2)WRITE(6,200)
IF(KEY.EQ.3)WRITE(6,300)
IF(KEY.EQ.3)WRITE(6,250)
DO 10 I=1,NOFLD
NV=FLDSAV(4,I)
NQ=NV-1
NR=NQ-5
IF(NQ.GT.5)NQ=5
IE=IR+NQ-1
IC=FLDSAV(2,I)
IS=FLDSAV(3,I)
FLDNAM=FLDSAV(1,I)
IF(KEY.EQ.3)GO TO 5
WRITE(6,700)I,FLDNAM,(LPRN,VERTEX(1,J),VERTEX(2,J),J=IB,IE)
IF(IS.EQ.1)WRITE(6,705)
IF(IS.EQ.2)WRITE(6,710)
GO TO 6
5 CONTINUE
IF(IS.EQ.0)WRITE(6,400)I,FLDNAM,IC,CLSNAM(IC),
* (LPRN,VERTEX(1,J),VERTEX(2,J),J=IR,IE)
IF(IS.NE.0)WRITE(6,500)I,FLDNAM,IC,CLSNAM(IC),IS,SUBNAM(IS),
* (LPRN,VERTEX(1,J),VERTEX(2,J),J=IB,IE)
6 CONTINUE
IF(NR.IE.0)GO TO 7
IR=IE+1
IE=IR+NR-1
WRITE(6,650)(LPRN,VERTEX(1,J),VERTEX(2,J),J=IB,IE)
7 CONTINUE
IR=IE+2
10 CONTINUE
RETURN
100 FORMAT(/// 20X,'AREA USED TO COMPUTE TRAINING STATISTICS'//)
200 FORMAT(/// 50X,'INPUT FIELDS'//)
250 FORMAT(/// 45X,'DESIGNATED FIELDS'///T18,'FIELD',T40,'DESIGNATED',
* TPO,'VERTICES (SAMPLE,LINE)'//)
300 FORMAT(1X,T18,'FIELD',T34,'CLASS',T47,'SUBCLASS',T80,'VERTICES (S
*AMPLE,LINE)'//)
400 FORMAT(T15,I3,T20,A4,T30,I3,T35,A4,T65,5(A1,I4,'.',I4,''),1X))
500 FORMAT(T15,I3,T20,A4,T30,I3,T35,A4,T45,I3,T50,A4,T65,
* 5(A1,I4,'.',I4,''),1X))
650 FORMAT(1X,T65,5(A1,I4,'.',I4,''),1X))
700 FORMAT(T15,I3,T20,A4,T65,5(A1,I4,'.',I4,''),1X))
705 FORMAT(1H+,T40,'UNIDENTIFIABLE')
710 FORMAT(1H+,T40,'OTHER')
END

```

WRT00010
WRT00020
WRT00030
WRT00040
WRT00050
WRT00060
WRT00070
WRT00080
WRT00090
WRT00100
WRT00110
WRT00120
WRT00130
WRT00140
WRT00150
WRT00160
WRT00170
WRT00180
WRT00190
WRT00200
WRT00210
WRT00220
WRT00230
WRT00240
WRT00250
WRT00260
WRT00270
WRT00280
WRT00290
WRT00300
WRT00310
WRT00320
WRT00330
WRT00340
WRT00350
WRT00360
WRT00370
WRT00380
WRT00390
WRT00400
WRT00410
WRT00420
WRT00430
WRT00440
WRT00450
WRT00460
WRT00470
WRT00480
WRT00490
WRT00500
WRT00510
WRT00520
WRT00530
WRT00540
WRT00550
WRT00560
WRT00570
WRT00580
WRT00590
WRT00600
WRT00610
WRT00620
WRT00630
WRT00640

FILE WRTHED

```

C*****
C SURROUTINE WRTHED(NCHAN,FEAT,NSAMP,FORMAT,IUNIT)
C IMPLICIT INTEGER (A-Z)
C*****
C THE PURPOSE OF TAPWRT IS TO WRITE A DATA TAPE IN EITHER UNIVER-
C SAL FORMAT OR LARSYS II FORMAT. THERE ARE TWO ENTRY POINTS TO
C THE SUBROUTINE -- WRTHED AND WRTLN.
C WRTHED WRITES THE HEADER RECORD IN 32 BIT BYTES FOR LARSYS II
C AND 8 BIT BYTES FOR UNIVERSAL. ONE CALL TO WRTHED MUST BE MADE
C FOR EACH REEL OF TAPE. THIS INFORMATION IS PACKED.
C
C CALL WRTHED(INC,FEAT,NSAMP,FORMAT,TRFORM)
C   NC -- NO. OF CHANNELS TO BE WRITTEN FOR EACH DATA SET
C   FEAT -- ARRAY CONTAINING CHANNELS TO BE WRITTEN
C   NSAMP -- NO. OF SAMPLES PER CHANNEL
C   FORMAT -- =1 FOR UNIVERSAL
C   TRFORM -- NO. OF TAPE OUTPUT UNIT
C           =2 FOR LARSYS II
C WRTLN WRITES THE DATA IN 8 BIT BYTES AND IS ALSO PACKED. A
C CALL TO THIS ROUTINE MUST BE MADE FOR EACH DATA SET TO BE WRIT-
C TEN
C
C CALL WRTLN (IDATA,LSTLIN)
C   IDATA -- ARRAY CONTAINING DATA TO BE WRITTEN
C   LSTLIN -- = 0 FOR N-1 DATA SETS
C             =-1 FOR LAST DATA SET
C
C   ICHAN -- ACTIVE CHANNELS HAVE CORRESPONDING BIT POSITION
C           TURNED ON
C
C   PROFLG --
C
C CONTINUE
C   NCS -- NO. OF CHANNELS
C   NRITS -- NO. OF BITS PER BYTE
C   SVD -- START OF VIDEO DATA
C   NVE -- SAME AS NSAMP
C   PRSZ -- PHYSICAL RECORD SIZE IN BYTES
C   NCPR -- NO. CHANNELS PER RECORD
C   NPPR -- NO. PHYSICAL RECORDS PER CHANNEL
C   NRPOS -- NO. OF RECORDS PER DATA SET
C   ANCLNG -- LENGTH OF ANCILLARY BLOCK IN BYTES
C   DOI -- DATA ORDER INDICATOR
C   SAMSTR -- SAMPLE START
C   COMWRD -- SIZE OF COMPUTER WORD IN BITS
C   NDSPR -- NO. OF DATA SETS PER RECORD
C   NCAP -- NO. OF CHANNELS ON ANCILLARY RECORD
C
C   PACRAY -- DATA IS PACKED INTO THIS ARRAY AND THEN WRITTEN ON
C             TAPE BY CALLING NTRAN
C   ICOUNT -- RUNNING TOTAL OF NO. OF DATA SETS WRITTEN
C*****
C LOGICAL*1 PACRAY(3060)
C REAL RAY(200)
C DIMENSION NR(18),PACK(765)
C DIMENSION IHYES(18),FEAT(30),IRAY(200)
C LOGICAL*1 VARIAB(2400)
C
C*** COMMON BLOCK CREATED AUG. 3,1979 TO SAVE LARSYS III HEADER
C
C COMMON /IDSTOR/ IOD(250)
C COMMON /WRTAP/ICOUNT,FORMAT,UNIT,VARBL(600),IREMD
C COMMON /TAPFRD/IUNIT7,IFRST,FSCAN,SAMEND,SAMINC,READY,
C 1 NSCAN,LINC,IO(200),OSL,LHUF(30),JREC(30),IHYTE(30),
C 2 NRUES,FILENO,LINEND,LININC,NSAMPZ,NCHAN7,FORMATZ
C DATA IHYES/81,89,90,91,92,96,100,102,103,104,105,107,108,110,
C 753,1778,1785,1787/
C DATA NR/4,1,1,1,2,2,2,1,1,1,2,1,2,2,1,1,2,2/
C EQUIVALENCE (VARBL,VARIAB),(PACK,PACRAY),(IRAY,RAY)
C EQUIVALENCE (VARBL(1),ICHAN), (VARBL(7),PRSZ), (VARBL(14),NOSAM),
C 1 (VARBL(2),PROFLG), (VARBL(8),NCPR), (VARBL(15),COMWRD)
C 2 (VARBL(3),NC), (VARBL(9),NPPR),
C 3 (VARBL(4),NRITS), (VARBL(10),NRPOS), (VARBL(16),NDSPR),
C 4 (VARBL(5),SVD), (VARBL(11),ANCLNG),
C 5 (VARBL(6),NVE), (VARBL(12),DOI), (VARBL(17),NCAP),
C 6 (VARBL(13),SAMSTR), (VARBL(18),NSAM)
C 7

```

```

NC = NCHAN
FORMAT = FRMAT
UNIT = IUNIT
ICOUNT=0
SAMSTR=1
  ICHAN = 0
  ISTAT = 0
  NVE = NSAMP
  NSAM = NSAMP
  NOSAM = NSAMP
  IF ( FORMT .EQ. 1) GO TO 40
ZERO OUT HEADER RECORD STORAGE
DO 5 I=1,200
  IRAY(I) = IDO(I)
PACKING HEADER RECORD IN LARSYS II FORMAT
  IRAY(5) =NC
  IRAY(6) =NSAMP + 6
  NORYTE = 800
  CALL WRTREC(UNIT,NORYTE,IRAY)
  RETURN
PACKING HEADER RECORD IN UNIVERSAL FORMAT
40 DO 50 I=1,NC
  II = FEAT(II)
50 ICHAN = ICHAN + 2**(32-II)
  NPRC = 0
  PROFLG = 1
  NBITS = 8
  SVD = 1
  PRSZ = 3060
  PRSZHD = 3060
  ANCLNG = 70
  ANC = ANCLNG + 2
  DOI = 0
  COMWRD = 32
  NDSPR = 1
  NCPR = 0
  IF (NSAMP .GT. 2998) WRITE(6,510)
510 FORMAT(' NO. OF SAMPLES WAS RESET TO 2998')
  IF (NSAMP .GT. 2998) NSAMP = 2998
  ILEN = (NC*NSAMP) + ANC
  ILENTN = (NC*NSAMP + ANC ) / 3000
  IREMD = MOD((NC*NSAMP + ANC ),3000)
  IF (ILENTN .EQ. 0) NCAR = NC
  IF (ILENTN .EQ. 0) NRPD = 1
  IF (ILENTN .EQ. 0) GO TO 80
  DO 60 J=1,NC
  IF((NSAMP*J+ANC ) .GT. 3000) NCAR = J-1
  IF((NSAMP*J+ANC ) .GT. 3000) GO TO 70
60 CONTINUE
70 CONTINUE
  NOCHAN = NC - NCAR
  DO 75 J=1,NOCHAN
  IF(NSAMP*J .GT. 2998) NCPR = J-1
  IF(NSAMP*J .GT. 2998) GO TO 76
75 CONTINUE
  NCPR = NOCHAN
76 CONTINUE
  NRPD = NOCHAN / NCPR + 1
  IF (MOD(NOCHAN,NCPR) .NE. 0) NRPD = NRPD + 1
80 CONTINUE
  IF (ILENTN.NE.0) GO TO 82
  PRSZ = (ILEN/180)*180
  IF (PRSZ.NE.ILEN) PRSZ =PRSZ + 180
82 CONTINUE
ZERO OUT PACRAY
DO 85 K=1,765
85 PACK(K) = 0

```

FILE WRTHEO

```

DO 100 K=1,18
NBYTES = NH(K)
DO 90 L=1,NBYTES
LOC = 4 * L - NBYTES + (K-1)*4
BYTE = IBYTES(K)
90 PACRAY(BYTE+L-1) = VARIAB(LOC)
100 CONTINUE
PACRAY(61) = VARIAB(73)
PACRAY(62) = VARIAB(74)
PACRAY(63) = VARIAB(75)
PACRAY(67) = VARIAB(79)
PACRAY(68) = VARIAB(80)
DO 110 L = 1,512
110 PACRAY(111 + L) = VARIAB(111 + L)
DO 120 L = 1,16
120 PACRAY(2200 + L) = VARIAB(2200 + L)
DO 130 I = 1,4
IPAT = 2254 + (I - 1)*8
130 PACRAY(IPAT) = VARIAB(IPAT)
C
C*** THE ABOVE THREE LINES IS AN AD HOC ADDITION FOR SOIL LINES
C*** ADDED OCT. 23,1978
C
CALL WRTREC(UNIT,PRSZHD,PACRAY)
RETURN
END

```

WRT01530
WRT01540
WRT01550
WRT01560
WRT01570
WRT01580
WRT01590
WRT01600
WRT01610
WRT01620
WRT01630
WRT01640
WRT01650
WRT01660
WRT01670
WRT01680
WRT01690
WRT01700
WRT01710
WRT01720
WRT01730
WRT01740
WRT01750
WRT01760
WRT01770
WRT01780

FILE WRTLN

```

C      SURROUTINE WRTLN(/IDATA/,LSTLIN)
      IMPLICIT INTEGER (A-Z)
      COMMON /WRTAP/ICOUNT,FORMAT,UNIT,VARBL(600),IREMD
      LOGICAL*1 PACRAY(13500),ISCAN(4),IDATA(1),IRECNO(4)
      LOGICAL*1 ZERO(4),LONE(4)
      EQUIVALENCE (LONE,IONE)
      DATA IONE/ZFFFF/
      EQUIVALENCE (ICOUNT,ISCAN)
      EQUIVALENCE (IRECNO,IRECNO)
      EQUIVALENCE (ZERO,IZERO)
      EQUIVALENCE (VARBL(3),NC),(VARBL(7),PRSZ),
      *          (VARBL(8),NCPR),(VARBL(9),NPRC),
      *          (VARBL(10),NRPDS),(VARBL(11),ANCLNG),
      *          (VARBL(18),NSAMP),(VARBL(17),NCAR)
      ICOUNT = ICOUNT + 1
      IZERO = 0
      ANC = ANCLNG + 2
      IF (FORMAT.EQ. 1)      GO TO 140

C      WRITES PACKED DATA ON TAPE IN LARSYS II FORMAT
C      650 FORMAT(3I5)
C      PACKING ONE SET OF DATA INTO ONE RECORD
C
      NRTS = 8
      ANCLNG = 4
      NRYTES = (NSAMP + 6)*NC
      PACRAY(1) = ISCAN(3)
      PACRAY(2) = ISCAN(4)
      PACRAY(3) = LONE(4)
      PACRAY(4) = LONE(4)

C*** ADDED AUG 10,1979 TO ADD CALIBRATION SPACE
C
      IV = 0
      III = 0
      DO 120 II = 1,NC
      DO 110 I = 1,NSAMP
      III = III + 1
      IV = IV + 1
      PACRAY(IV + 4) = IDATA(III*4)
110      CONTINUE
      IV = IV + 6
120      CONTINUE
      NRYTES=NRYTES+ANCLNG
      IDUM=(NRYTES/4)*4
      IIDUM=NRYTES-IDUM
      IF (IIDUM.NE.0) NRYTES=NRYTES+4-IIDUM
      CALL WRTREC(UNIT,NRYTES,PACRAY)
      IF (LSTLIN.EQ. -1) ENDFILE UNIT
      RETURN

C      WRITE PACKED DATA ON TAPE IN UNIVERSAL FORMAT
C
140 DO 150 I=1,72
150 PACRAY(I) = ZERO(4)
      ROW = 1
      NR = NRPDS - 1
      IF (NCPR.EQ.0) GO TO 155
      NCLR = MOD((NC-NCAR),NCPR)
155 WORD = 73

C      PACKING ANCILLARY INFORMATION INTO PACRAY
C
      RECNO = 1
      PACRAY(2) = IRECNO(4)
      PACRAY(71) = ISCAN(3)
      PACRAY(72) = ISCAN(4)

C      DATA IS NOT PACKED WITH ANCILLARY RECORD
C
      IF (NCAR.NE. 0)      GO TO 160
      NRYTES = NSAMP * NC
      KA = 1
      GO TO 210

```

WRT00010
 WRT00020
 WRT00030
 WRT00040
 WRT00050
 WRT00060
 WRT00070
 WRT00080
 WRT00090
 WRT00100
 WRT00110
 WRT00120
 WRT00130
 WRT00140
 WRT00150
 WRT00160
 WRT00170
 WRT00180
 WRT00190
 WRT00200
 WRT00210
 WRT00220
 WRT00230
 WRT00240
 WRT00250
 WRT00260
 WRT00270
 WRT00280
 WRT00290
 WRT00300
 WRT00310
 WRT00320
 WRT00330
 WRT00340
 WRT00350
 WRT00360
 WRT00370
 WRT00380
 WRT00390
 WRT00400
 WRT00410
 WRT00420
 WRT00430
 WRT00440
 WRT00450
 WRT00460
 WRT00470
 WRT00480
 WRT00490
 WRT00500
 WRT00510
 WRT00520
 WRT00530
 WRT00540
 WRT00550
 WRT00560
 WRT00570
 WRT00580
 WRT00590
 WRT00600
 WRT00610
 WRT00620
 WRT00630
 WRT00640
 WRT00650
 WRT00660
 WRT00670
 WRT00680
 WRT00690
 WRT00700
 WRT00710
 WRT00720
 WRT00730
 WRT00740
 WRT00750
 WRT00760

FILE WRTLN

```

C
C 160 ALL DATA IS PACKED ON ANCILLARY RECORD
      IF (NCAR .NE. NC) GO TO 170
      NBYTES = IREMD - ANC
      KA = 2
      GO TO 210
C
C 170 PART OF DATA IS PACKED ON ANCILLARY RECORD
      NBYTES = NCAR*NSAMP
      KA = 3
      GO TO 210
C
C 180 DATA IS PACKED ON MORE THAN ONE RECORD
      ANC = 2
      WORD = 3
      KA = 4
      J = 0
      185 J = J + 1
          IF (J.GT.NR) GO TO 200
          WRITE(6,660)NR
        660 FORMAT('NR',I5)
          RECNO = RECNO + 1
          PACRAY(2) = IRECNO(4)
          NBYTES = NCPR * NSAMP
          IF (NCLR .NE. 0 .AND. J .EQ. NR) NBYTES = NCLR*NSAMP
          GO TO 210
        200 CONTINUE
          IF (LSTLIN .EQ. -1) ENDFILE UNIT
          RETURN
        210 IF (NCAR .EQ. 0) GO TO 220
          II = (ROW-1)*NSAMP*4
          DO 215 I=1,NBYTES
        215 PACRAY(WORD*I-1) = IDATA(4*I+II)
        220 CALL WRTREC(UNIT,PRSZ,PACRAY)
          IF (KA .NE. 4) ROW = NCAR + ROW
          IF (KA .EQ. 4) ROW = ROW + NCPR
          GO TO (180,200,180,185),KA
      END

```

WRT00770
 WRT00780
 WRT00790
 WRT00800
 WRT00810
 WRT00820
 WRT00830
 WRT00840
 WRT00850
 WRT00860
 WRT00870
 WRT00880
 WRT00890
 WRT00900
 WRT00910
 WRT00920
 WRT00930
 WRT00940
 WRT00950
 WRT00960
 WRT00970
 WRT00980
 WRT00990
 WRT01000
 WRT01010
 WRT01020
 WRT01030
 WRT01040
 WRT01050
 WRT01060
 WRT01070
 WRT01080
 WRT01090
 WRT01100
 WRT01110
 WRT01120
 WRT01130
 WRT01140
 WRT01150
 WRT01160

FILE: WRTMTX

```

C      SUBROUTINE WRTMTX(MATICE,SIZE,BCD)
C      IMPLICIT INTEGER (A-H,O-Z)
C      DIMENSION FORMAT(6)
C      REAL MATICE(1)
C-----
C      CALL..    CALL WRTMTX(MATICE,SIZE,FREQ,BCD,MAXFET)
C      ARGS..    MATICE - COVARIANCE MATRICE
C                SIZE  - RANK OF 'MATRICE' ('DMATIC')
C                FREQ  - FREQUENCY MATRIX
C                BCD   - CONTAINS BCD PRECISION FOR PRINTOUT
C                MAXFET - NUMBER OF FEATURES PER LINE
C
C      REQUIRES.. NONE
C      PURPOSE..  PRINTS THE SINGLE-PRECISION COVARIANCE MATRICES
C      RETURNS.. NO CHANGE
C-----
C      CONTINUE
C      CALL..    CALL DWRTMX(MATICE,SIZE,FREQ,BCD,MAXFET)
C      ARGS..    SEE ABOVE
C      PURPOSE.. PRINTS THE DOUBLE-PRECISION COVARIANCE MATRICES
C      RETURNS.. SEE ABOVE
C-----
C-----
C      DATA FORMAT/'(1H0','1,6X','12F9','1,1' '1,1)' '/'
C      DOUPRE = 0
C      GO TO 10
C      ENTRY DWRTMX(DMATIC,SIZE,BCD)
C      DOUBLE PRECISION DMATIC(1)
C      DOUPRE = 1
C 10  FORMAT(5)=BCD
C      DO 100 LOC=1,SIZE,12
C      STOP = LOC+11
C      IF ( STOP .GT. SIZE) STOP = SIZE
C      II = 1
C      KINC = 1
C      DO 90 I=LOC,SIZE
C      K = I*(I+1)/2-II+1
C      JK = K-KINC+1
C      IF(DOUPRE.EQ.0) WRITE(6,FORMAT) (MATICE(J),J=K,JK)
C      IF(DOUPRE.EQ.1) WRITE(6,FORMAT) (DMATIC(J),J=K,JK)
C      II = II+1
C 90  IF(KINC.LT.12.AND.KINC.LT.STOP)KINC=KINC+1
C      WRITE(6,1004)
C 100 CONTINUE
C 1004 FORMAT('0')
C      RETURN
C-----
C      END

```

WRT00010
 WRT00020
 WRT00030
 WRT00040
 WRT00050
 WRT00060
 WRT00070
 WRT00080
 WRT00090
 WRT00100
 WRT00110
 WRT00120
 WRT00130
 WRT00140
 WRT00150
 WRT00160
 WRT00170
 WRT00180
 WRT00190
 WRT00200
 WRT00210
 WRT00220
 WRT00230
 WRT00240
 WRT00250
 WRT00260
 WRT00270
 WRT00280
 WRT00290
 WRT00300
 WRT00310
 WRT00320
 WRT00330
 WRT00340
 WRT00350
 WRT00360
 WRT00370
 WRT00380
 WRT00390
 WRT00400
 WRT00410
 WRT00420
 WRT00430
 WRT00440
 WRT00450
 WRT00460
 WRT00470
 WRT00480
 WRT00490
 WRT00500
 WRT00510
 WRT00520
 WRT00530
 WRT00540
 WRT00550
 WRT00560
 WRT00570
 WRT00580
 WRT00590
 WRT00600
 WRT00610
 WRT00620
 WRT00630
 WRT00640
 WRT00650
 WRT00660
 WRT00670
 WRT00680
 WRT00690

FILE WRTREC

```

SUBROUTINE WRTREC(UNIT,LENGTH,IBUF)
IMPLICIT INTEGER (A-Z)
C
C   OUTPUTS A SCAN LINE OF DATA
C
C   DIMENSION IAUF(3000)
C   LENGTH = LENGTH/4
C   WRITE(6,200)UNIT,LENGTH,IAUF(18)
C   WRITE(UNIT,100)((IBUF(I),I=1,LENGTH)
C   WRITE(6,200)UNIT,LENGTH,IBUF(18)
200  FORMAT(7I6)
100  FORMAT(3I(250A4))
RETURN
END

```

```

WRT00010
WRT00020
WRT00030
WRT00040
WRT00050
WRT00060
WRT00070
WRT00080
WRT00090
WRT00100
WRT00110
WRT00120
WRT00130
WRT00140

```


FILE DAMRG

~~20-1~~
482

FILE DAMRG

```

C
75  NUMFIL=1
   GO TO 100
C
   PROCESSING SECOND FILE FOR LARSYS III INTERLEAVED
C
   IUNIT=IDATTP(2)
90  READ(IUNIT,1100,END=100)DUMMY
1100 FORMAT(1A4)
   GO TO 90
C*** NUMBER OF CHANNELS FOR THIS FILE
C
100  NF = NFEAT(I)
   CALL FLDINT(FLDINF(1,I),FETVEC(1,I),NF)
C
C*** SET FEATURE COUNTER
C
   IF(I.GT.1) NI = NI + NFEAT(I - 1)
C
C*** SET SCALAR FIELD DESCRIPTION FOR THIS FILE
C
   SAMSTR = FLDINF(4,I)
   SAMINC = FLDINF(5,I)
   SAMEND = FLDINF(6,I)
   LINSTR = FLDINF(1,I)
   LININC = FLDINF(3,I)
   LINEND = FLDINF(2,I)
C
C*** NO. SAMPLES/LINE FOR FILE I COMPUTED IN FLDINT
C
   NS = NSAMP
C
C*** TOTAL NUMBER OF RADIANCE VALUES PER LINE
   NV = NS * NF
C
C*** WRITE FIELD INFORMATION FOR FILE I
C
   IF(SWITCH.EQ.1.AND.I.EQ.2)GO TO 1090
   WRITE (PRTUNT,1060) I
1060  FORMAT('O INPUT FIELD DESCRIPTION FOR FILE',I8)
   WRITE (PRTUNT,1070)
1070  FORMAT('O START LINE END LINE LINE INC START PIXEL END
   *PIXEL PIXEL INC')
   IF(SWITCH.EQ.1)GO TO 1085
   WRITE (PRTUNT,1080) (FLDINF(J,I), J=1,6)
1080  FORMAT('O 1.6(4X,14.4X))
   GO TO 1090
1085  WRITE(PRTUNT,1080)FLDSTR,FLDLST,(FLDINF(J,I),J=3,6)
C
C*** STORE SUN ANGLES
C
1090  IF(ISUNT.NE.0)GO TO 610
   IF(ISUNC.EQ.0)GO TO 620
C
C*** SUN ANGLES FROM CARDS
C
   DO 600 J = 1,8
   ID(16 + J) = ISUN(J,I)
   SUNANG(J) = ISUN(J,I)
600  CONTINUE
   GO TO 620
C
C*** SUN ANGLES FROM TAPE HEADER
C
610  DO 615 J = 1,8
   ISUN(J,I) = SUNANG(J)
615  CONTINUE
   CALL SUNFAC(SUNCOR,SUNANG,FETVEC(1,I),NF,ISUNT,ISUNC)
620  CONTINUE
   IF (FORMM.NF.1) GO TO 660
   WRITE (PRTUNT,1000) I
1000  FORMAT('O DATA FOR INPUT FILE',I4)
   WRITE (PRTUNT,1010)IDL(100),IDL(104),IDL(108),IDL(111),IDL(112)
1010  FORMAT('O INPUT FILE DATE AND SITE',5(3X,Z2))
C
C*** LOAD VARJAB WITH EXTRA HEADER INFORMATION

```

DAM00770
 DAM00780
 DAM00790
 DAM00800
 DAM00810
 DAM00820
 DAM00830
 DAM00840
 DAM00850
 DAM00860
 DAM00870
 DAM00880
 DAM00890
 DAM00900
 DAM00910
 DAM00920
 DAM00930
 DAM00940
 DAM00950
 DAM00960
 DAM00970
 DAM00980
 DAM00990
 DAM01000
 DAM01010
 DAM01020
 DAM01030
 DAM01040
 DAM01050
 DAM01060
 DAM01070
 DAM01080
 DAM01090
 DAM01100
 DAM01110
 DAM01120
 DAM01130
 DAM01140
 DAM01150
 DAM01160
 DAM01170
 DAM01180
 DAM01190
 DAM01200
 DAM01210
 DAM01220
 DAM01230
 DAM01240
 DAM01250
 DAM01260
 DAM01270
 DAM01280
 DAM01290
 DAM01300
 DAM01310
 DAM01320
 DAM01330
 DAM01340
 DAM01350
 DAM01360
 DAM01370
 DAM01380
 DAM01390
 DAM01400
 DAM01410
 DAM01420
 DAM01430
 DAM01440
 DAM01450
 DAM01460
 DAM01470
 DAM01480
 DAM01490
 DAM01500
 DAM01510
 DAM01520

FILE DAMRG

```

C
C*** LOAD DATE AND SITE FROM FIRST FILE
C
    IF(I.NE.1)GO TO 625
    VARIAB(73) = IDL(100)
    VARIAB(74) = IDL(104)
    VARIAB(75) = IDL(108)
    VARIAB(79) = IDL(111)
    VARIAB(80) = IDL(112)

C
C*** FOR PURPOSES OF UNIVERSAL HEADER WRITE LOAD VARIAB WITH SUN ANGLES
C*** AND GAINS AND PHASES ONLY IF CHANNEL MERGE OPTION
C
    625 IF(IMOPT.NE.1)GO TO 660
    DO 650 J = 1,NF
    IDUM = (FETVEC(J,1) - 1) * 2
    I1 = 112 + IDUM
    I2 = 112 + N1 * 2 + (J - 1) * 2
    VARIAB(I2) = IDL(I1 + 3)
    VARIAB(I2 + 1) = IDL(I1 + 4)
    I1 = 240 + IDUM
    I2 = 240 + N1 * 2 + (J - 1) * 2
    VARIAB(I2) = IDL(I1 + 3)
    VARIAB(I2 + 1) = IDL(I1 + 4)
    I1 = 368 + IDUM
    I2 = 368 + N1 * 2 + (J - 1) * 2
    VARIAB(I2) = IDL(I1 + 3)
    VARIAB(I2 + 1) = IDL(I1 + 4)
    I1 = 496 + IDUM
    I2 = 496 + N1 * 2 + (J - 1) * 2
    VARIAB(I2) = IDL(I1 + 3)
    VARIAB(I2 + 1) = IDL(I1 + 4)
    650 CONTINUE
    KS = 0
    DO 655 J = 1,NF
    IDUM = FETVEC(J,1)
    IDUM = (IDUM - 1) / NCHPAS
    IF(ISUNT.EQ.0.AND.J.EQ.1) KS = IDUM
    I1 = (IDUM - KS) * 4 + 3
    I2 = 2201 + (N1 + J - 1) * 2
    VARIAB(I2) = LOGSUN(I1)
    VARIAB(I2 + 1) = LOGSUN(I1 + 1)
    655 CONTINUE
    660 CONTINUE

C
C*** THE NEXT LINE WAS ADDED OCT. 23,1978 AS AN AD HOC ADDITION
C*** TO ADD SOIL LINES TO THE UNIVERSAL HEADER
C
    VARIAB(2246 + 8*I) = IDL(640)

C
C
C*** INITIALIZATION FOR LINE EXTRACTION PARAMETERS NEEDED FOR
C*** SPATIAL MERGE
C
    LOC = (I - 1) / NACROS
    N5 = 0
    IF (LOC.EQ.0) GO TO 666
    DO 661 J = 1,LOC
    N5 = N5 + NLINES(J)
    661 CONTINUE
    666 LREM = (I - 1) - LOC * NACROS
    N2 = 0
    IF (LREM.EQ.0)GO TO 663
    DO 662 J = 1,LREM
    N2 = N2 + NSS(J)
    662 CONTINUE
    663 JCT = 0

C
C*** PARAMETERS NEEDED IF PSEUDO MERGE OPTION
C
    LPTR = LINPTR(I)
    NL = NLINES(I)
    NLM = NL + LPTR - 1

C
C*** EXTRACT FIELD FOR THIS FILE LINE BY LINE
C

```

DAM01530
 DAM01540
 DAM01550
 DAM01560
 DAM01570
 DAM01580
 DAM01590
 DAM01600
 DAM01610
 DAM01620
 DAM01630
 DAM01640
 DAM01650
 DAM01660
 DAM01670
 DAM01680
 DAM01690
 DAM01700
 DAM01710
 DAM01720
 DAM01730
 DAM01740
 DAM01750
 DAM01760
 DAM01770
 DAM01780
 DAM01790
 DAM01800
 DAM01810
 DAM01820
 DAM01830
 DAM01840
 DAM01850
 DAM01860
 DAM01870
 DAM01880
 DAM01890
 DAM01900
 DAM01910
 DAM01920
 DAM01930
 DAM01940
 DAM01950
 DAM01960
 DAM01970
 DAM01980
 DAM01990
 DAM02000
 DAM02010
 DAM02020
 DAM02030
 DAM02040
 DAM02050
 DAM02060
 DAM02070
 DAM02080
 DAM02090
 DAM02100
 DAM02110
 DAM02120
 DAM02130
 DAM02140
 DAM02150
 DAM02160
 DAM02170
 DAM02180
 DAM02190
 DAM02200
 DAM02210
 DAM02220
 DAM02230
 DAM02240
 DAM02250
 DAM02260
 DAM02270
 DAM02280

FILE DAMRG

```

DO 690 II = LINSTR,LINEND,LININC
ICT = ICT + 1
CALL LINERD(ARRAY(1),ENDTAP)
IF (ENDTAP.EQ.-1) CALL CMERR
IF (IMOPT.NE.3) GO TO 670
C*** LOOK FOR LINE MATCH IF PSEUDO MERGE
C
DO 665 J = LPTR,NLM
IF (II.EQ.LINES(J)) GO TO 670
665 CONTINUE
GO TO 690
670 CONTINUE
ICCT = ICCT + 1
C
IF (IPPPP.EQ.1) WRITE(PRTUNT,1020) (ARRAY(K),K=1,NV)
1020 FORMAT(/10I7)
IF (IMOPT.NE.1) GO TO 675
C*** CHANNEL MERGE MODE WRITE NV VALUES TO DIRECT ACCESS FILE
C
IF (ISOPT.EQ.0) GO TO 672
C*** DO SUN ANGLE CORRECTION
C
DO 671 J = 1,NF
DO 671 JJ = 1,NS
ITEMP = (JJ + (J - 1) * NS)
DUM = SUNCOR(J) * FLOAT(ARRAY(ITEMP))
ARRAY(ITEMP) = IFIX(DUM)
671 CONTINUE
672 ADDRES = DRUMAD + N1*NS + (ICT - 1)*NS*NOFEAT
CALL RWRITE(ADDRES,ARRAY(1),NV,STATUS)
GO TO 690
675 IF (IMOPT.NE.2) GO TO 680
C*** SPATIAL MERGE MODE WRITE NSS(I)*NF VALUES TO DIRECT ACCESS FILE
C
IF (ISOPT.EQ.0) GO TO 677
DO 676 J = 1,NF
DO 676 JJ = 1,NS
ITEMP = (JJ + (J - 1) * NS)
DUM = SUNCOR(J) * FLOAT(ARRAY(ITEMP))
ARRAY(ITEMP) = IFIX(DUM)
676 CONTINUE
677 N4 = NSS(I)
DO 679 J = 1,NF
ADDRES = DRUMAD + (NS + ICT-1)*NOSAMP*NF + NOSAMP*(J-1)*N2
ITEMP = 1 + (J-1)*N4
CALL RWRITE(ADDRES,ARRAY(ITEMP),N4,STATUS)
679 CONTINUE
GO TO 690
C*** PSEUDO MERGE OPTION
C
680 IF (ISOPT.EQ.0) GO TO 682
DO 681 J = 1,NF
DO 681 JJ = 1,NS
ITEMP = (JJ + (J - 1) * NS)
DUM = SUNCOR(J) * FLOAT(ARRAY(ITEMP))
ARRAY(ITEMP) = IFIX(DUM)
681 CONTINUE
682 ADDRES = DRUMAD + (ICCT - 1) * NV
CALL RWRITE(ADDRES,ARRAY(1),NV,STATUS)
690 CONTINUE
C*** LINE LOOP COMPLETE
C
700 CONTINUE
C*** LOOP FOR FILE I COMPLETE
C*** WRITE OUTPUT FILE
C
DATFI = DATFIL - 1
C

```

DAM02290
DAM02300
DAM02310
DAM02320
DAM02330
DAM02340
DAM02350
DAM02360
DAM02370
DAM02380
DAM02390
DAM02400
DAM02410
DAM02420
DAM02430
DAM02440
DAM02450
DAM02460
DAM02470
DAM02480
DAM02490
DAM02500
DAM02510
DAM02520
DAM02530
DAM02540
DAM02550
DAM02560
DAM02570
DAM02580
DAM02590
DAM02600
DAM02610
DAM02620
DAM02630
DAM02640
DAM02650
DAM02660
DAM02670
DAM02680
DAM02690
DAM02700
DAM02710
DAM02720
DAM02730
DAM02740
DAM02750
DAM02760
DAM02770
DAM02780
DAM02790
DAM02800
DAM02810
DAM02820
DAM02830
DAM02840
DAM02850
DAM02860
DAM02870
DAM02880
DAM02890
DAM02900
DAM02910
DAM02920
DAM02930
DAM02940
DAM02950
DAM02960
DAM02970
DAM02980
DAM02990
DAM03000
DAM03010
DAM03020
DAM03030
DAM03040

*** POSITION OUTPUT FILE

C

WIND DATA

```
CALL FSFMFL (DATAPE,DATFI,ISTAT)
```

c

C*** SET OUTPUT CHANNELS 1,2,...,NOFEAT

C

DO R00 I = 1.NOFEAT

FETVEC(I,1) = I

800 CONTINUE

9

```
C*** WRITE HEADER OF OUTPUT FILE
```

C

CALL WRTHED(NOFEAT,FETVEC(1,1),NOSAMP,FORMM,DATAP)

C

C*** EXTRACT SCAN LINES ONE AT A TIME WRITE TO OUTPUT FILE

2

LSTLIN = 0

NV = NOSAMP * NOFEAT

NO 850 I = 1. NOLINE

```
IF (I.EQ.NOLINE) LSTLIN = - 1
```

ADDRESS = DRUMAD + (I - 1) * NV

CALL RREAD(ADDRESS,ARRAY(1),NV,ISTAT)

CALL WRTLN (ARRAY(1),LSTLIN)

```
1030 IF (IPPPP.EQ.1) WRITE (PRINT,1030) I
      FORMAT ('OUTPUT LINE:',I6)
```

```
FORMAT('DDDDPPT LINE V.16')  
IF(LPPPP-FA-1)WRITE(OUTUN
```

```
850 CONTINUE
```

850 CONTINUE

c.

C*** RETURN TO MONITOR

c

2

*** OUTPUT FILE COMPLETED

c

RETURN

END

DAMO3050
DAMO3060
DAMO3070
DAMO3080
DAMO3090
DAMO3100
DAMO3110
DAMO3120
DAMO3130
DAMO3140
DAMO3150
DAMO3160
DAMO3170
DAMO3180
DAMO3190
DAMO3200
DAMO3210
DAMO3220
DAMO3230
DAMO3240
DAMO3250
DAMO3260
DAMO3270
DAMO3280
DAMO3290
DAMO3300
DAMO3310
DAMO3320
DAMO3330
DAMO3340
DAMO3350
DAMO3360
DAMO3370
DAMO3380
DAMO3390
DAMO3400

```

SUBROUTINE SET1A(ARRAY,TOP)
IMPLICIT INTEGER(A-Z)
COMMON/GLOBAL/HEAD(63),MAPTAP,DATAP,SAVTAP,BMFILE,BMKEY,
*HISFIL,HISKEY,TIFORM,ERIPTP,ERPKEY,MAPUNT,NOFILE,
*DRUMAD,DRMDS,PAGSIZ,DATFIL,STAFIL,ASAV,ASAVFL,
*NHSTUN,NHSTFI,SCTRUN,MAMPFIL,
*DOTUNT,DOTFIL,NCHPAS,TRNSFL,RMTREFL,HISTFL,PCHUNT,
*CRDUNT,PRTUNT,RANDIO
COMMON/TAPERD/IUNIT,IFRST,FSCAN,SAMEND,SAMINC,READY,NSCAN,
*LINC,ID(200),DSL,LHUF(30),JREC(30),IRYTE(30),NRUFS,FILENO,LINEND,
*LININC,NSAMP,NCHAN,FORMAT
COMMON/WRTAP/ICOUNT,DUMMY,UNIT,VARBL(600),IREMD
COMMON/SOLNA/SUNANG(A),ISUNT,ISUNC,SMSTR,SMINC,LINSKP
COMMON/MRGDAI/IMOPT,ISOPT,NUMFIL,IDATT(6),IDATFL(6),
*NOFEAT,NFEAT(6),FETVEC(30,6),ISUN(8,6),SUNCOR(30),
*FLDNF(6,6),NOSAMP,NCLINE,NSS(6),NACROS,NLINES(6),LINPTR(7),
*LINES(600),FORMM
DIMENSION ARRAY(1)
DIMENSION HED1(15),HED2(15),DATE(3),COMENT(15),EQUVEC(2),SLASH(2)
EQUIVALENCE (HED1,HEAD(4)),(DATE(1),HEAD(22)),(HED2(1),HEAD(30)),
*(COMENT(1),HEAD(48))
DATA EQUVEC/1,'='/,SLASH/1,'/'/,
DATA IRCD/'I'/',OBOD/'O'/',BLANK/' ',UBCD/'U'/',
*LRCD/'L'/',CRCD/'C'/',SRCD/'S'/',PBCD/'P'/',
*ARCD/'A'/',TRCD/'T'/',FRCD/'F'/'
DIMENSION INVE(13)
DATA INVE/'NCPA','FORM','DATE','HED1','HED2','NLIN','OPTI',
*NACR','SUNA','DATA','LINE','CHAN','*END'/
DIMENSION CARD(62),ACARD(20)

C*** CHANNEL CARD COUNTER
ICHNCT = 0

C*** SUN ANGLE COUNTER - CONTROL CARDS
ISUNCT = 0

C*** INPUT DATA TAPE COUNTER
NUMFIL = 0
NOLINE = 0

C*** DEFAULT SETTINGS
IMOPT = 1
ISUNT = 0
ISOPT = 0
ISUNC = 0
NACROS = 1
IDATT(1) = 22
IDATFL(1) = 1
NOFEAT = 4
DO 10 J = 1,6
DO 10 J = 1,4
NFEAT(J) = 4
FETVEC(J,I) = J
10 CONTINUE
DO 20 I = 1,6
DO 20 J = 1,8
ISUN(J,I) = 60
20 CONTINUE
FORMM = 1
PRUNIT = 30
REWIND PRUNIT
READ(CRDUNT,1000)(ACARD(I),I = 1,20)
1000 FORMAT(20A4)
WRITE(PRUNIT,1000)(ACARD(I),I = 1,20)
REWIND PRUNIT
READ(RRUNIT,1010)CODE,CARD
1010 FORMAT(A4,6X,62A1)
REWIND RRUNIT
WRITE(PRTUNT,1020)CODE,CARD
1020 FORMAT(7X,A4,6X,62A1)
COL = 0
ISTART = 0
DO 90 I = 1,13
IF(CODE.EQ.INVE(I))GO TO(100,150,200,250,300,350,400,450,500,
*550,600,650,700).I

```

FILE: SET18

```

      90 CONTINUE
      WRITE(PRTUNT,1030)CODE,CARD
1030  FORMAT(' INVALID CARD - IGNORED'//T5,A4,6X,62A1)
      GO TO 80
C*** NUMBER OF CHANNELS PER PASS CARD IMAGE
C
100  J = NXTCHR(CARD,COL)
      IF(J.EQ.BLANK)GO TO 80
      J = NUMRER(CARD,COL,NCHPAS,ISTART)
      GO TO 80
C*** FORMAT CARD IMAGE - OUTPUT FILE
C
150  J = NXTCHR(CARD,COL)
      IF(J.EQ.LRCD) FOKMM = 2
      GO TO 80
C*** DATE CARD
C
200  READ(RRUNIT,1040)DATE
1040  FORMAT(10X,3A4)
      REWIND RRUNIT
      GO TO 80
C*** HED1 CARD
C
250  READ(RRUNIT,1050)HED1
1050  FORMAT(10A,15A4)
      REWIND RRUNIT
      GO TO 80
C*** HED2 CARD
C
300  READ(RRUNIT,1050)HED2
      REWIND RRUNIT
      GO TO 80
C*** NLINES (6) CARD FOR PSEUDO OPTION
C
350  J = NUMBER(CARD,COL,ARRAY(1),ISTART)
      IF(J.GT.6) J = 6
      DO 360 JJ = 1,J
360  NLINFS(JJ) = ARRAY(JJ)
      GO TO 80
C*** OPTION CARD IMAGE
C
400  J = NXTCHR(CARD,COL)
      IF(J.EQ.CRCD) IMOPT = 1
      IF(J.EQ.SRCD) IMOPT = 2
      IF(J.EQ.PRCD) IMOPT = 3
      IF(J.EQ.ABCD) ISOPT = 1
      GO TO 80
C*** SPATIAL OPTION ... NUMBER FIELDS TO BE JOINED ACROSS
C
450  J = NUMRER(CARD,COL,NACROS,ISTART)
      GO TO 80
C*** SUN ANGLE CARD IMAGES
C
500  J = NXTCHR(CARD,COL)
      IF(J.EQ.IRCD)GO TO 510
      ISUNT = 1
      GO TO 80
510  ISUNCT = ISUNCT + 1
      COL = 0
      ISUNC = 1
      J = NUMBER(CARD,COL,ARRAY(1),ISTART)
      DO 520 JJ = 1,J
      ISUN(JJ,ISUNCT) = ARRAY(JJ)
520  CONTINUE
      GO TO 80
C*** DATA TAPE CARD IMAGES
C
550  J = NXTCHR(CARD,COL)
      IF(J.EQ.IHCD)GO TO 570
```

SET00800
SET00810
SET00820
SET00830
SET00840
SET00850
SET00860
SET00870
SET00880
SET00890
SET00900
SET00910
SET00920
SET00930
SET00940
SET00950
SET00960
SET00970
SET00980
SET00990
SET01000
SET01010
SET01020
SET01030
SET01040
SET01050
SET01060
SET01070
SET01080
SET01090
SET01100
SET01110
SET01120
SET01130
SET01140
SET01150
SET01160
SET01170
SET01180
SET01190
SET01200
SET01210
SET01220
SET01230
SET01240
SET01250
SET01260
SET01270
SET01280
SET01290
SET01300
SET01310
SET01320
SET01330
SET01340
SET01350
SET01360
SET01370
SET01380
SET01390
SET01400
SET01410
SET01420
SET01430
SET01440
SET01450
SET01460
SET01470
SET01480
SET01490
SET01500
SET01510
SET01520
SET01530
SET01540
SET01550
SET01560
SET01570
SET01580

FILE: SET18

| | | |
|------|---|----------|
| C | IF(J.NE.ORCD)GO TO 595 | SET01590 |
| C*** | OUTPUT FILE | SET01600 |
| C | J = FIND12(CARD,COL,SLASH) | SET01610 |
| | IF(J.NE.2)GO TO 595 | SET01620 |
| | J = NXTCHR(CARD,COL) | SET01630 |
| | IF(J.EQ.FRCD)GO TO 555 | SET01640 |
| C*** | UNIT NUMBER OF OUTPUT FILE, THEN FILE | SET01650 |
| C | J = FIND12(CARD,COL,EQUVEC) | SET01660 |
| | IF(J.NE.2)GO TO 595 | SET01670 |
| | J = NUMBER(CARD,COL,DATAPE,ISTART) | SET01680 |
| | J = FIND12(CARD,COL,EQUVEC) | SET01690 |
| | IF(J.NE.2)GO TO 595 | SET01700 |
| | ISTART = 0 | SET01710 |
| | J = NUMBER(CARD,COL,DATFIL,ISTART) | SET01720 |
| | GO TO A0 | SET01730 |
| C*** | FILE NUMBER, THEN UNIT | SET01740 |
| C | 555 J = FIND12(CARD,COL,EQUVEC) | SET01750 |
| | IF(J.NE.2)GO TO 595 | SET01760 |
| | J = NUMBER(CARD,COL,DATFIL,ISTART) | SET01770 |
| | J = FIND12(CARD,COL,EQUVEC) | SET01780 |
| | IF(J.NE.2)GO TO 595 | SET01790 |
| | ISTART = 0 | SET01800 |
| | J = NUMBER(CARD,COL,DATAPE,ISTART) | SET01810 |
| | GO TO A0 | SET01820 |
| C*** | INPUT FILES | SET01830 |
| C | 570 NUMFIL = NUMFIL + 1 | SET01840 |
| | IF(NUMFIL.GT.6)GO TO 590 | SET01850 |
| | J = FIND12(CARD,COL,SLASH) | SET01860 |
| | IF(J.NE.2)GO TO 590 | SET01870 |
| | J = NXTCHR(CARD,COL) | SET01880 |
| | IF(J.EQ.FRCD)GO TO 575 | SET01890 |
| C*** | UNIT NUMBER OF INPUT FILE, THEN FILE NUMBER | SET01900 |
| C | J = FIND12(CARD,COL,EQUVEC) | SET01910 |
| | IF(J.NE.2)GO TO 590 | SET01920 |
| | J = NUMBER(CARD,COL,IDATTP(NUMFIL),ISTART) | SET01930 |
| | J = FIND12(CARD,COL,EQUVEC) | SET01940 |
| | IF(J.NE.2)GO TO 590 | SET01950 |
| | ISTART = 0 | SET01960 |
| | J = NUMBER(CARD,COL,IDATFL(NUMFIL),ISTART) | SET01970 |
| | GO TO A0 | SET01980 |
| 575 | J = FIND12(CARD,COL,EQUVEC) | SET01990 |
| | IF(J.NE.2)GO TO 590 | SET02000 |
| | J = NUMBER(CARD,COL,IDATFL(NUMFIL),ISTART) | SET02010 |
| | J = FIND12(CARD,COL,EQUVEC) | SET02020 |
| | IF(J.NE.2)GO TO 590 | SET02030 |
| | ISTART = 0 | SET02040 |
| | J = NUMBER(CARD,COL,IDATTP(NUMFIL),ISTART) | SET02050 |
| | GO TO A0 | SET02060 |
| C*** | ERRORS | SET02070 |
| C | 590 NUMFIL = NUMFIL - 1 | SET02080 |
| | 595 WRITE(PRTUNT,1060) | SET02090 |
| | 1060 FORMAT(' ERROR ON ABOVE INPUT CONTROL CARD') | SET02100 |
| | GO TO A0 | SET02110 |
| C*** | LINES IN PSEUDO OPTION | SET02120 |
| C | 600 NOLINE = NUMBER(CARD,COL,LINES,NOLINE) | SET02130 |
| | GO TO A0 | SET02140 |
| C*** | CHANNELS CARD | SET02150 |
| C | 650 ICHNCT = ICHNCT + 1 | SET02160 |
| | IF(ICHNCT.GT.6)GO TO A0 | SET02170 |
| | J = NUMBER(CARD,COL,ARRAY(1),ISTART) | SET02180 |
| | IF(J.GT.30)J = 30 | SET02190 |
| | DO 660 JJ = 1,J | SET02200 |
| | FETVEC(JJ,ICHNCT) = ARRAY(JJ) | SET02210 |
| | | SET02220 |
| | | SET02230 |
| | | SET02240 |
| | | SET02250 |
| | | SET02260 |
| | | SET02270 |
| | | SET02280 |
| | | SET02290 |
| | | SET02300 |
| | | SET02310 |
| | | SET02320 |
| | | SET02330 |
| | | SET02340 |
| | | SET02350 |
| | | SET02360 |
| | | SET02370 |

FILE: SET18

```
660 CONTINUE
NFEAT(ICHNCT) = J
GO TO 80
C
C*** *END CARD
C
700 LIMIT = 1
IF(IMOPT.EQ.2)LIMIT = NUMFIL
DO 710 J = 1,LIMIT
ARRAY(2) = 0
JJ = LAREAD(ARRAY(1),ARRAY(3),FLDINF(1,J),ARRAY(2))
JSAVE = J
IF(JJ.NE.1)GO TO 750
710 CONTINUE
JJ = LAREAD(ARRAY(1),ARRAY(3),ARRAY(4),ARRAY(2))
IF(JJ.NE.0)GO TO 750
GO TO 770
C
C*** ERROR IN FIELD CARDS
C
750 WRITE(PRTUNT,1070)JSAVE
1070 FORMAT(' ERROR IN FIELD',I10,' OR SEND CARD MISSING')
CALL CMERR
C
C*** PROCESS INFORMATION
C
770 CONTINUE
IF(ICHNCT.EQ.NUMFIL)GO TO 775
WRITE(PRTUNT,1080)ICHNCT,NUMFIL
1080 FORMAT(' NUMBER OF CHANNEL CARDS',I5,' DOES NOT MATCH NUMBER OF ',
*DATA FILES',I5)
ICHNCT = NUMFIL
775 IF(ISUNC.EQ.1.AND.ISUNCT.NE.NUMFIL)GO TO 780
GO TO 785
780 WRITE(PRTUNT,1090)ISUNCT,NUMFIL
1090 FORMAT(' NUMBER OF SUN ANGLE CARDS',I5,' DOES NOT MATCH NUMBER',
*OF DATA FILES',I5)
ISUNCT = NUMFIL
785 IF(IMOPT.EQ.1)GO TO 795
C
C*** CHECK NO. CHANNELS EQUAL ON SPATIAL OR PSEUDO OPTION
C
NOFEAT = NFEAT(1)
IF (NUMFIL.EQ.1) GO TO 795
DO 790 J = 2,NUMFIL
IF(NFEAT(J).EQ.NFEAT(1))GO TO 790
WRITE(PRTUNT,1100)J,NFEAT(J),NFEAT(1)
1100 FORMAT(' NUMBER OF FEATURES OF',I5,' FILE',I5,
*IS NOT EQUAL TO FIRST',I5)
NFEAT(J) = NFEAT(1)
790 CONTINUE
GO TO 806
C
C*** FEATURES IN CHANNEL MERGE
C
795 IDUM = 0
DO 800 I = 1,NUMFIL
IDUM = IDUM + NFEAT(I)
800 CONTINUE
IF(IDUM.LE.30)GO TO 805
WRITE(PRTUNT,1105)(NFEAT(I),I = 1,NUMFIL)
1105 FORMAT(' FEATURES ADD UP TO A NUMBER GREATER THAN 30',6I5,
*EXITING')
CALL CMERR
805 NOFEAT = IDUM
C
C*** SET NOSAMP AND NOLINE
C
806 IF (IMOPT.NE.2) NACROS = 1
NOSAMP = 0
DO 810 J = 1,NACROS
N55(J) = (FLDINF(5,J) - FLDINF(4,J))/FLDINF(6,J) + 1
NOSAMP = NOSAMP + N55(J)
810 CONTINUE
NDOWN = NUMFIL / NACROS
IF(IMOPT.NE.2)NDOWN = 1
NOLINE = 0
ICT = 0
NDOWN1 = NDOWN*NACROS
```

SET02390
SET02390
SET02400
SET02410
SET02420
SET02430
SET02440
SET02450
SET02460
SET02470
SET02480
SET02490
SET02500
SET02510
SET02520
SET02530
SET02540
SET02550
SET02560
SET02570
SET02580
SET02590
SET02600
SET02610
SET02620
SET02630
SET02640
SET02650
SET02660
SET02670
SET02680
SET02690
SET02700
SET02710
SET02720
SET02730
SET02740
SET02750
SET02760
SET02770
SET02780
SET02790
SET02800
SET02810
SET02820
SET02830
SET02840
SET02850
SET02860
SET02870
SET02880
SET02890
SET02900
SET02910
SET02920
SET02930
SET02940
SET02950
SET02960
SET02970
SET02980
SET02990
SET03000
SET03010
SET03020
SET03030
SET03040
SET03050
SET03060
SET03070
SET03080
SET03090
SET03100
SET03110
SET03120
SET03130
SET03140
SET03150
SET03160

FILE: SET1A

```
      DO A15 J = 1,NDOWN1,NACROS
      ICT = ICT + 1
      IDUM = (FLDINF(2,J) - FLDINF(1,J))/FLDINF(3,J) + 1
      NOLINE = NOLINE + IDUM
      IF (IMOPT.EQ.2) NLINES(ICT) = IDUM
A15  CONTINUE
      IF(NDOWN.EQ.1)GO TO 830
      NDOWN1 = NDOWN - 1
      DO A20 J = 1,NDOWN1
      DO A20 JJ = 1,NACROS
      ITEMF = JJ + J * NACROS
      NSS(ITEMF) = NSS(JJ)
A20  CONTINUE
C*** SET LINPTR
C
C A30 LINPTR(1) = 1
C
C*** STORE FLDINF
C
      IF(IMOPT.EQ.2)GO TO 833
      DO A32 J = 2,6
      DO A32 JJ = 1,6
      FLDINF(JJ,J) = FLDINF(JJ,1)
A32  CONTINUE
A33  IF(IMOPT.NE.3)GO TO 840
      DO A35 J = 1,NUMFIL
      LINPTR(J + 1) = LINPTR(J) + NLINES(J)
A35  CONTINUE
A40  RETURN
      END
```

SET03170
SET031A0
SET03190
SET03200
SET03210
SET03220
SET03230
SET03240
SET03250
SET03260
SET03270
SET03280
SET03290
SET03300
SET03310
SET03320
SET03330
SET03340
SET03350
SET03360
SET03370
SET03380
SET03390
SET03400
SET03410
SET03420
SET03430
SET03440
SET03450
SET03460
SET03470

21. GTDDM PROCESSOR

FILE: GTDDM

```
C      WRITTEN BY C W AHLERS
C      GROUND TRUTH TAPE DUMP ROUTINE
C      SUBROUTINE GTDDM(ARRAY, TOP)
      IMPLICIT INTEGER (A-Z)
      DIMENSION ARRAY(1)
      CALL SET14
      CALL DDM(ARRAY, TOP)
      RETURN
      END
```

```
GTDD00010
GTDD00020
GTDD00030
GTDD00040
GTDD00050
GTDD00060
GTDD00070
GTDD00080
GTDD00090
GTDD00100
GTDD00110
```

FILE: ALPHA

```
FUNCTION ALPHA(S)
  IMPLICIT INTEGER (A-Z)
  DIMENSION A(26)
  DATA A/'A','B','C','D','E','F','G','H','I','J','K','L','M',
    'N','O','P','Q','R','S','T','U','V','W','X','Y','Z'/
  DO 10 I=1,26
    ALPHA=I
  IF (S.EQ.A(I)) RETURN
10  CONTINUE
  WRITE(6,20) S
20  FORMAT(1H,5X,'THE SYMBOL ',A1,' CAN NOT BE USED.')
  RETURN
END
```

ALP00010
ALP00020
ALP00030
ALP00040
ALP00050
ALP00060
ALP00070
ALP00080
ALP00090
ALP00100
ALP00110
ALP00120
ALP00130

ORIGINAL PAGE 1
OF POOR QUALITY

~~21-2~~
493

FILE: NDM

```
C      WRITTEN BY C W AHLERS                                DDM00010
      SUBROUTINE NDM (ARRAY, TOP)                            DDM00020
      IMPLICIT INTEGER (A-Z)                                DDM00030
      DIMENSION ARRAY(1)                                     DDM00040
      DIMENSION NMTX(11,19)                                 DDM00050
      DIMENSION FETVEC(30)                                   DDM00060
      DATA FETVEC/30*0/                                     DDM00070
      DIMENSION IDATA(3060)                                  DDM00080
      DIMENSION RLOCK(6)                                     DDM00090
      COMMON /TAPEPD/ IUNIT, IFIRST, FSCAN, SAMEND, SAMINC, READY, NSCAN,
      * LINE, ID(200), DSL, LRUF(30), JREC(30), IBYTE(30), NBUFS, FILENO, LINEND
      * LININC, NSAMP, NOCHAN, IFORMT                        DDM00110
      * COMMON /GTRK/NRDR, NPRT, PRKEY, VLB(6), GTRDU, GTRDF,
      * GTWRU, GTWRF, GTNOF                                  DDM00120
      * GTWRU, GTWRF, GTNOF                                  DDM00130
      * GTWRU, GTWRF, GTNOF                                  DDM00140
      * GTWRU, GTWRF, GTNOF                                  DDM00150
      EQUIVALENCE (IDATA(1), ARRAY(1))                       DDM00160
      EQUIVALENCE (IDATA(1), ARRAY(3061))                   DDM00170
      UNIT=GTWRU                                              DDM00180
      IUNIT=GTWRU                                              DDM00190
      FORMT=IFORMT                                             DDM00200
      NFOF=0                                                  DDM00210
      FILE=0                                                  DDM00220
      NOFFAT=NOCHAN                                           DDM00230
      FETVEC(1)=1                                             DDM00240
      DAY=0                                                    DDM00250
      MON=0                                                    DDM00260
      YEAR=0                                                  DDM00270
      SITE=0                                                  DDM00280
      PX=196                                                  DDM00290
      RLOCK(1)=1                                              DDM00300
      RLOCK(2)=117                                           DDM00310
      RLOCK(3)=1                                              DDM00320
      RLOCK(4)=1                                              DDM00330
      RLOCK(5)=196                                           DDM00340
      RLOCK(6)=1                                              DDM00350
10      CONTINUE                                             DDM00360
      REWIND GTRDU                                             DDM00370
      REWIND GTWRU                                             DDM00380
      RDF=GTRDF-1                                             DDM00390
      WRF=GTWRF-1                                             DDM00400
C 10      CONTINUE                                             DDM00410
      NRPOS=1                                                 DDM00420
      MAXREC=3060                                             DDM00430
      ICOUNT=0                                                DDM00440
      NREC=0                                                  DDM00450
      FILE=FILE+1                                             DDM00460
      LINE=0                                                  DDM00470
      WRITE (NPRT, 951)                                       DDM00480
      FORMAT(1H1)                                             DDM00490
      CALL TAPHDR(GTRDU, RDF)                                  DDM00500
      CALL FSEFPL(GTWRU, WRF, ISTATW)                        DDM00510
      WRITE (NPRT, 601) GTRDF, GTWRF                         DDM00520
601      FORMAT(//, ' TAPE FILE ', I4, 5X, ' BEING DUMPED TO DOT FILE ', I4)
      DAY=ID(25)                                              DDM00530
      MON=ID(26)                                              DDM00540
      YEAR=ID(27)                                             DDM00550
      SITE=ID(28)                                             DDM00560
      WRITE (NPRT, 306) SITE, DAY, MON, YEAR                 DDM00570
      IF (PRKEY.EQ.1) WRITE (GTWRU, 356) SITE, DAY, MON, YEAR
      IF (PRKEY.EQ.2) WRITE (GTWRU, 366) SITE, DAY, MON, YEAR
      IF (PRKEY.EQ.3) WRITE (GTWRU, 376) SITE, DAY, MON, YEAR
306      FORMAT(' SITE = ', I5, 5X, ' DAY = ', I5, 5X, ' MONTH = ', I5, 5X,
      * ' YEAR = ', I5)
356      FORMAT(' SITE = ', I5, 1X, ' DAY = ', I5, 1X, ' MONTH = ', I5, 1X,
      * ' YEAR = ', I5, ' MASK = TY ', I, ' TYPE = GT ')
366      FORMAT(' SITE = ', I5, 1X, ' DAY = ', I5, 1X, ' MONTH = ', I5, 1X,
      * ' YEAR = ', I5, ' MASK = PHASH 3 ', I, ' TYPE = GT ')
376      FORMAT(' SITE = ', I5, 1X, ' DAY = ', I5, 1X, ' MONTH = ', I5, 1X,
      * ' YEAR = ', I5, ' MASK = INPUT ', I, ' TYPE = GT ')
      WRITE (NPRT, 600)
      CALL FLDIR (RLOCK, FETVEC, NOFFAT)
600      FORMAT(///, ' THE 200 OUT LABELS ')
20      CONTINUE
      IF (NREC.EQ. RLOCK(2)) GO TO 50
      CALL LTRFND (IDATA, ENDTAP)
      IF (ENDTAP.EQ.-1) GO TO 55
      NFOF=0
      NREC=NREC+1
C      IF (NREC.LT.2) WRITE (NPRT, 850) (IDATA(K), K=1, 63)
      DDM00700
      DDM00710
      DDM00720
      DDM00730
      DDM00740
      DDM00750
      DDM00760
      DDM00770
      DDM00780
      DDM00790
```

FILE: DDM

```

850  FORMAT(1H,1514)
      N10=MOD(NREC,10)
C    IF (NREC.EQ.1) WRITE(NPRT,850) (IDATA(K),K=1,GTREC)
      IF (N10.EQ.0) WRITE(NPRT,836)
      • (IDATA(1),1=10,140,10)
836  FORMAT(1H,5X,1915)
      IF (N10.EQ.0) CALL GTDOTS
      • (IDATA,DMTX,LINE)
839  FORMAT(1H,5X,1915)
      GO TO 20
50   CONTINUE
      CALL GTDTL(DMTX,NSYM)
      TYPE=PRKEY
      CALL GTDNW(DMTX,TYPE,NSYM)
      WRITE(NPRT,700) NREC
700  FORMAT(///,' NUMREK OF SCAN LINES READ=',15)
      WRITE(NPRT,750) GTDOF,GTWRF
750  FORMAT(' FILE ',15,5X,' WAS DUMPED TO FILE ',15)
      GTDOF=GTDOF+1
      GTWRF=GTWRF+1
      DOF=DOF+1
      WRF=WRF+1
      ENDFILE GTWRU
      IF (FILE,LT,GTNOF) GO TO 10
55   CONTINUE
      REWIND GTIRU
      REWIND GTWRU
      WRITE(NPRT,250)
250  FORMAT(' PROGRAM GTDDM RUN COMPLETED')
      RETURN
      END

```

DDM00800
DDM00810
DDM00820
DDM00830
DDM00840
DDM00850
DDM00860
DDM00870
DDM00880
DDM00890
DDM00900
DDM00910
DDM00920
DDM00930
DDM00940
DDM00950
DDM00960
DDM00970
DDM00980
DDM00990
DDM01000
DDM01010
DDM01020
DDM01030
DDM01040
DDM01050
DDM01060
DDM01070
DDM01080
DDM01090
DDM01100

FILE: GTDOTS

```
      SUBROUTINE GTDOTS(/IDATA/,DMTX,LINE)
      IMPLICIT INTEGER (A-Z)
      COMMON /GT-K/NROW,NPRT,PRTKEY,VLB(6),GTRDU,
1         GTRIF,GTRHU,GTRPF,GTRNF
      DIMENSION DMTX(11,19),IDATA(1)
C 901  WRITE(6,901)
      FORMAT(1H,10X,'GTDOTS')
      LINE=LINE+1
      DO 10 I=10,190,10
      K=I/10
      DMTX(LINE,K)=IDATA(I)
10  CONTINUE
      RETURN
      END
```

GTD00010
GTD00020
GTD00030
GTD00040
GTD00050
GTD00060
GTD00070
GTD00080
GTD00090
GTD00100
GTD00110
GTD00120
GTD00130
GTD00140

THIS PAGE IS
TY

FILE: GTOTL

| | | |
|-----|--|----------|
| C | WRITTEN C W AHLENS | GT000010 |
| | SUBROUTINE GTOTL(DMTX,NSYM) | GT000020 |
| | IMPLICIT INTEGER (A-Z) | GT000030 |
| | COMMON /TR/ TRNS1(256),TRNS2(26),TRNS3(26),TY(11,19) | GT000040 |
| | DATA B/1,1/ | GT000050 |
| | DIMENSION DMTX(11,19) | GT000060 |
| | DO 5 I=1,26 | GT000070 |
| | TRNS2(I)=B | GT000080 |
| | TRNS3(I)=B | GT000090 |
| 5 | CONTINUE | GT000100 |
| | DO 10 I=1,11 | GT000110 |
| | DO 10 J=1,19 | GT000120 |
| | GD=DMTX(I,J) | GT000130 |
| | SYM=TRNS1(GD) | GT000140 |
| | NUM=ALPHA(SYM) | GT000150 |
| | TRNS2(NUM)=SYM | GT000160 |
| | DMTX(I,J)=SYM | GT000170 |
| 10 | CONTINUE | GT000180 |
| C | WRITE(6,40) (TRNS2(I),I=1,26) | GT000190 |
| 60 | FORMAT(1H,5X,26A1) | GT000200 |
| | NSYM=0 | GT000210 |
| | DO 100 I=1,26 | GT000220 |
| | IF (TRNS2(I).NE.B) NSYM=NSYM+1 | GT000230 |
| | IF (TRNS2(I).NE.B) TRNS3(NSYM)=TRNS2(I) | GT000240 |
| 100 | CONTINUE | GT000250 |
| | WRITE(6,110) (TRNS3(I),I=1,26) | GT000260 |
| 110 | FORMAT(///,5X,'CATEGORIES FOUND: ',26A1) | GT000270 |
| | WRITE(6,20) | GT000280 |
| 20 | FORMAT(1H,///,5X,'THE 209 DOTS TRANSFORMED') | GT000290 |
| | DO 40 I=1,11 | GT000300 |
| | WRITE(6,30) (DMTX(I,J),J=1,19) | GT000310 |
| 40 | CONTINUE | GT000320 |
| 30 | FORMAT(1H,5X,19(4X,A1)) | GT000330 |
| | RETURN | GT000340 |
| | END | GT000350 |

FILE GTDWR

| | | |
|-----|--|----------|
| C | WRITTEN BY C W AHLERS | GTD00010 |
| C | THIS PROGRAM WRITES LACIE FORMAT DOT FILES | GTD00020 |
| C | GIVEN A MATRIX OF DOT LABELS & A TYPE MASK MATRIX | GTD00030 |
| | SUBROUTINE GTDWR(DMTX,TYPE,NSYM) | GTD00040 |
| | IMPLICIT INTEGER (A-Z) | GTD00050 |
| | DIMENSION DMTX(11,19) | GTD00060 |
| | DIMENSION DWR(15) | GTD00070 |
| | COMMON /TR/TRANS1(256),TRANS2(26),TRANS3(26),TY(11,19) | GTD00080 |
| | COMMON /GT4K/NRDH,NPRT,PRTKEY,VLB(6),GTRDU,GTRDF, | GTD00090 |
| | * GTWRU,GTWRF,GTNOF | GTD00100 |
| | WRITE(NPRT,11) | GTD00110 |
| 11 | FORMAT(1H,/,10X,'LACIE FORMAT DOT LABELS') | GTD00120 |
| | WRITE(NPRT,111) GTWRU | GTD00130 |
| 111 | FORMAT(1H,10X,'WRITTEN TO UNIT ',I5) | GTD00140 |
| | IF(TYPE.EQ.1) WRITE(NPRT,12) | GTD00150 |
| | IF(TYPE.EQ.2) WRITE(NPRT,13) | GTD00160 |
| | IF(TYPE.EQ.3) WRITE(NPRT,14) | GTD00170 |
| 12 | FORMAT(/,10X,'TYPES BASED ON A TRANSITION YEAR MASK') | GTD00180 |
| 13 | FORMAT(/,10X,'TYPES BASED ON A PHASE THREE MASK') | GTD00190 |
| 14 | FORMAT(/,10X,'TYPES BASED ON AN INPUT MASK') | GTD00200 |
| | NTYPES=2 | GTD00210 |
| | DO 30 TT=1,NTYPES | GTD00220 |
| | DO 30 SS=1,NSYM | GTD00230 |
| | COUNT=0 | GTD00240 |
| | DO 25 I=1,11 | GTD00250 |
| | DO 25 J=1,19 | GTD00260 |
| | IJ=J+19*(I-1) | GTD00270 |
| | IF(TY(I,J).EQ.TT.AND.DMTX(I,J).EQ.TRNS3(SS)) COUNT=COUNT+1 | GTD00280 |
| | IF(TY(I,J).EQ.TT.AND.DMTX(I,J).EQ.TRNS3(SS)) DWR(COUNT)=IJ | GTD00290 |
| | IF(COUNT.EQ.15) WRITE(NPRT,33) TT,TRANS3(SS),(DWR(K),K=1,COUNT) | GTD00300 |
| | IF(COUNT.EQ.15) WRITE(GTWRU,43) TT,TRANS3(SS),(DWR(K),K=1,COUNT) | GTD00310 |
| | IF(COUNT.EQ.15) COUNT=0 | GTD00320 |
| 25 | CONTINUE | GTD00330 |
| | IF(COUNT.GT.0) WRITE(NPRT,33) TT,TRANS3(SS),(DWR(K),K=1,COUNT) | GTD00340 |
| | IF(COUNT.GT.0) WRITE(GTWRU,43) TT,TRANS3(SS),(DWR(K),K=1,COUNT) | GTD00350 |
| 33 | FORMAT(1H,5X,'DOT ',I1,2X,A1,3X,15I4) | GTD00360 |
| 43 | FORMAT('DOT ',I1,2X,A1,3X,15I4) | GTD00370 |
| 30 | CONTINUE | GTD00380 |
| | WRITE(GTWRU,53) | GTD00390 |
| 53 | FORMAT('\$END') | GTD00400 |
| | RETURN | GTD00410 |
| | END | GTD00420 |

ORIGINAL PAGE IS
OF POOR QUALITY

FILE: GTRNS

```
      SUBROUTINE GTRNS"
      IMPLICIT INTEGER (A-Z)
      DATA W,S,R,F,O,N/1W,1S,1R,1F,1O,1N/
      COMMON /TR/TRANS1(256),TRANS2(256),TRANS3(256)
      DO 10 I=1,256
        TRANS1(I)=N
10      CONTINUE
        TRANS1(99)=W
        TRANS1(124)=W
        TRANS1(100)=S
        TRANS1(125)=S
        TRANS1(101)=R
        TRANS1(126)=R
        TRANS1(102)=R
        TRANS1(127)=R
        TRANS1(103)=F
        TRANS1(128)=F
        TRANS1(104)=O
        TRANS1(129)=O
      DO 20 I=1,15
        II=I+15
        TRANS1(I)=W
        TRANS1(II)=S
20      CONTINUE
      RETURN
      END
```

GTT00010
GTT00020
GTT00030
GTT00040
GTT00050
GTT00060
GTT00070
GTT00080
GTT00090
GTT00100
GTT00110
GTT00120
GTT00130
GTT00140
GTT00150
GTT00160
GTT00170
GTT00180
GTT00190
GTT00200
GTT00210
GTT00220
GTT00230
GTT00240
GTT00250
GTT00260

FILE: SET19

ORIGINAL PART 1
OF POOR QUALITY

```

C      WRITTEN BY C W AHLERS
SUBROUTINE SET19
IMPLICIT INTEGER (A-Z)
DIMENSION CODE(10),CARD(62),EQUOM(3),ACARD(20)
DIMENSION SLASH(2)
DATA SLASH /1,1//
DATA CODE/TRAN,READ,WRITE,
* MASK,DATE,COMM,HED1,HED2,END,DUMP/
DATA EQUOM/2,1,1//
DATA V/V//,PLNK//,U/U//,FF/F//,OO/O//,P/P//
DATA T/T//
DATA QQ/Q//,A,II/I//
C      INCLUDE COMRK1.LIST
C      INCLUDE COMRK4.LIST
C      INCLUDE COMRK6.LIST
C      INCLUDE COMRK14.LIST
COMMON/INFORM/NOCL52,NOSUR2,NOFET2,VARSZ2,TOTVT2,NOFLD2,
* AVAR2,CVAR2,CLSID2,SURN02,SURDS2,FLDSV2,VERTX2,
* FEIVC2(30),SUHVC2(75),SUHPT2(75),CLSV2(60),
* KEPT5(60),NOGRP,GRPNAM(60),GRPDEX(61),
* GRPCHK(61),GROUPS(124)
DIMENSION HED1(15),HED2(15),DATE(3),COMENT(15)
EQUIVALENCE (HED1(1),HEAD(4)),(DATE(1),HEAD(22)),
2 (HED2(1),HEAD(30)),(COMENT(1),HEAD(48))
COMMON/GLOBAL/HEAD(63),MAPTAP,DATEP,SAVTAP,BMFILE,BMKEY,
* HISFIL,HISKEY,TRFORM,ERIPTR,ERPKEY,MAPUNT,NOFILE,
* DRUMAD,DRUMDS,PAGSIZ,DATEFIL,STAFIL,ASAV,ASAVL
* ,NHSTUN,NHSTFI,SCRUN,MAPFIL
* ,DOTUNT,DOTFIL,NCHPAS,TRNSFL,RMTRFL,HISTFL,PCHUNT,
* CPDUNT,PRUNT,RANDIO
COMMON /GTRK/NRDR,NPRT,MSKKEY,VLR(6),GTRDU,GTRDF,
* GTWRU,GTWRF,GTNOF
COMMON /TAPERD/IUNIT,IFRST,FSANC,SAMEND,SAMINC,READY,NSCAN,
* LINC,LD(200),DSL,LHUF(30),JREC(30),IHYE(30),NBUFS,FILENO,LINEND
* ,LININC,NSAMP,NOCHAN,IFORMT
C$END COMMON /TR/TRANS1(256),TRANS2(26),TRANS3(26),TY(11,19)
DIMENSION PH31(19),PH32(19),PH33(19),PH34(19),PH35(19)
DIMENSION PH36(19),PH37(19),PH38(19),PH39(19),PH310(19)
DIMENSION PH311(19)
DATA PH31/5*1,2,2,1,1,2,2,8*1/
DATA PH32/1,2,1,2,1,2,2,1,2,2,1,2,1,1,2,1,1,2,1/
DATA PH33/1,2,2,3*1,2,1,2,3*1,3*2,1,2,1,2,1/
DATA PH34/2,2,1,1,2,1,2,2,3*1,2,1,2,2,1,2,2,2/
DATA PH35/1,2,2,3*1,3*2,1,1,2,1,1,2,2,3*1/
DATA PH36/1,2,2,1,4*2,1,1,4*2,1,2,1,2,1/
DATA PH37/1,2,2,1,2,3*1,2,2,1,2,1,3*2,1,1,2,1/
DATA PH38/3*1,10*2,1,4*2,1/
DATA PH39/1,2,1,2,1,1,2,2,1,1,2,1,2,1,4*2,1/
DATA PH310/4*1,4*2,1,1,2,3*1,2,1,1,2,2,1/
DATA PH311/1,2,1,1,2,5*1,2,1,1,2,2,3*1,2,1/
DIMENSION NVEC(30)
C
C
7FPO = 0
GOOF=0
GTRDU=12
GTWRU=23
GTRDF=1
GTWRF=1
GTNOF=1
LINE=0
NPRT=PRUNT
NPDR=CPDUNT
NOLAP = 0
MSKKEY = 1
NPJT=10
GTNOF=1
NOCHAN=1
IFORMT=1
DO 50 I=1,256
50 TRANS1(I)=00
C
DO 55 I=1,11
DO 55 J=1,19
TY(I,J)=-100
55 CONTINUE
WRITE(NPRT,100)

```

SET00010
SET00020
SET00030
SET00040
SET00050
SET00060
SET00070
SET00080
SET00090
SET00100
SET00110
SET00120
SET00130
SET00140
SET00150
SET00160
SET00170
SET00180
SET00190
SET00200
SET00210
SET00220
SET00230
SET00240
SET00250
SET00260
SET00270
SET00280
SET00290
SET00300
SET00310
SET00320
SET00330
SET00340
SET00350
SET00360
SET00370
SET00380
SET00390
SET00400
SET00410
SET00420
SET00430
SET00440
SET00450
SET00460
SET00470
SET00480
SET00490
SET00500
SET00510
SET00520
SET00530
SET00540
SET00550
SET00560
SET00570
SET00580
SET00590
SET00600
SET00610
SET00620
SET00630
SET00640
SET00650
SET00660
SET00670
SET00680
SET00690
SET00700
SET00710
SET00720
SET00730
SET00740
SET00750
SET00760
SET00770
SET00780
SET00790

FILE: SET10

```

100 FORMAT(/1X,'INPUT SUMMARY'//)
C
C   SET UP REREAD BUFFER
C
C   RRUNIT = 30
C   CALL REREAD(RRUNIT,80)
C
C   PUT CARD IN BUFFER
C
105 READ(NROR,103)(ACARD(I),I=1,20)
103 FORMAT(20A4)
WRITE(RRUNIT,103)(ACARD(I),I=1,20)
REWIND RRUNIT
C
READ(RRUNIT,110)CODE1,CARD
REWIND RRUNIT
COL = 0
WRITE(NPRT,120)CODE1,CARD
120 FORMAT(1X,A4,6X,62A1)
110 FORMAT(A4,6X,62A1)
DO 130 I=1,NPRT
IF (CODE1.EQ.CODE(I)) GO TO (150,180,210,330,370,
390,400,410,420,500),I
130 CONTINUE
135 WRITE(NPRT,140)
140 FORMAT(' INVALID CONTROL CARD - IGNORED ')
GO TO 105
C
C   TRANSFORMATION
C
150 M = NLTCHR(CARD,COL)
IF (M.EQ.BLNK) GO TO 105
GO TO 155
152 WRITE(NPRT,153)
153 FORMAT(' ERROR ON TRANSFORMATION CARDS')
GO TO 105
155 J = FIND12(CARD,COL,EQUOM)
IF (J.NE.2) GO TO 152
NOLAB=0
NOLAB = NUMBER(CARD,COL,VLR,NOLAB)
COL = COL - 1
IF (NOLAB.GT.2) GO TO 152
IF (VLR(1).GT.VLR(2)) GO TO 152
IF (VLR(2).GT.256) GO TO 152
NR=VLR(1)
NE=VLR(2)
DO 156 I=NR,NE
TRANS(I)=M
156 CONTINUE
GO TO 105
C
C   READ TAPE
C
180 M = NLTCHR(CARD,COL)
IF (M.EQ.BLNK) GO TO 105
IF (M.EQ.U) GO TO 190
IF (M.EQ.FF) GO TO 200
185 WRITE(NPRT,187)
187 FORMAT(' ERROR ON READ TAPE CARD')
GO TO 105
190 J = FIND12(CARD,COL,EQUOM)
IF (J.NE.2) GO TO 185
M = NUMBER(CARD,COL,GTRDU,ZERO)
COL = COL - 1
GO TO 180
200 J = FIND12(CARD,COL,EQUOM)
IF (J.NE.2) GO TO 185
M = NUMBER(CARD,COL,GTRDF,ZERO)
COL = COL - 1
GO TO 180
C
C   WRITE FILE
C
210 CONTINUE
214 M = NLTCHR(CARD,COL)
IF (M.EQ.BLNK) GO TO 105
IF (M.EQ.U) GO TO 230
IF (M.EQ.FF) GO TO 240
215 WRITE(NPRT,220)

```

```

SET00800
SET00810
SET00820
SET00830
SET00840
SET00850
SET00860
SET00870
SET00880
SET00890
SET00900
SET00910
SET00920
SET00930
SET00940
SET00950
SET00960
SET00970
SET00980
SET00990
SET01000
SET01010
SET01020
SET01030
SET01040
SET01050
SET01060
SET01070
SET01080
SET01090
SET01100
SET01110
SET01120
SET01130
SET01140
SET01150
SET01160
SET01170
SET01180
SET01190
SET01200
SET01210
SET01220
SET01230
SET01240
SET01250
SET01260
SET01270
SET01280
SET01290
SET01300
SET01310
SET01320
SET01330
SET01340
SET01350
SET01360
SET01370
SET01380
SET01390
SET01400
SET01410
SET01420
SET01430
SET01440
SET01450
SET01460
SET01470
SET01480
SET01490
SET01500
SET01510
SET01520
SET01530
SET01540
SET01550
SET01560
SET01570
SET01580

```

FILE: SET19

ORIGINAL PAGE 1
OF POOR QUALITY

```
220 FORMAT(' ERROR ON WRITE FILE CARD')
GO TO 105
230 J = FIND12(CARD,COL,EUCOM)
IF (J.NE.2) GO TO 215
M = NUMBER(CARD,COL,GTWRU,ZERO)
COL = COL - 1
GO TO 214
240 J = FIND12(CARD,COL,EUCOM)
IF (J.NE.2) GO TO 215
M = NUMBER(CARD,COL,GTWRF,ZERO)
COL = COL - 1
GO TO 214
CC
CC
CC
MASK CARD
330 M = NXTCHR(CARD,COL)
IF (M.EQ. BLNK) GO TO 105
IF (M.EQ.P) GO TO 340
IF (M.EQ.T) GO TO 341
IF (M.EQ.I) GO TO 342
333 WRITE(UNIT,335)
335 FORMAT(' ERROR ON MASK CARD --- TRANSITION YEAR MASK USED')
GO TO 341
340 MSKKEY = 2
DO 20 J=1,19
TY(1,J)=PH31(J)
TY(2,J)=PH32(J)
TY(3,J)=PH33(J)
TY(4,J)=PH34(J)
TY(5,J)=PH35(J)
TY(6,J)=PH36(J)
TY(7,J)=PH37(J)
TY(8,J)=PH38(J)
TY(9,J)=PH39(J)
TY(10,J)=PH310(J)
TY(11,J)=PH311(J)
20 CONTINUE
GO TO 105
341 MSKKEY=1
DO 10 I=1,11
DO 10 J=1,19
IJ=J+10*(I-1)
TY(I,J)=1
IF(IJ/2.EQ.IJ) TY(I,J)=2
10 CONTINUE
IF(GOOF.EQ.1) GO TO 461
GO TO 105
342 MSKKEY=3
LINE=LINE+1
IF(LINE.GT.11) GO TO 333
J=FIND12(CARD,COL,EUCOM)
IF(J.NE.2) GO TO 333
NOMSK=0
NOMSK=NUMBER(CARD,COL,MVEC,NOMSK)
IF(NOMSK.GT.19) GOOF=1
IF(NOMSK.GT.19) GO TO 105
DO 345 J=1,NOMSK
TY(LINE,J)=MVEC(J)
345 CONTINUE
GO TO 105
CC
CC
CC
DATE CARD
370 M = NXTCHR(CARD,COL)
IF (M.EQ. BLNK) GO TO 105
READ(UNIT,370) DATE
380 FORMAT(10X,15A4)
REWIND UNIT
GO TO 105
CC
CC
CC
COMMENT CARD
390 M = NXTCHR(CARD,COL)
IF (M.EQ. BLNK) GO TO 105
READ(UNIT,390) COMMENT
REWIND UNIT
GO TO 105
CC
CC
```

SET01590
SET01600
SET01610
SET01620
SET01630
SET01640
SET01650
SET01660
SET01670
SET01680
SET01690
SET01700
SET01710
SET01720
SET01730
SET01740
SET01750
SET01760
SET01770
SET01780
SET01790
SET01800
SET01810
SET01820
SET01830
SET01840
SET01850
SET01860
SET01870
SET01880
SET01890
SET01900
SET01910
SET01920
SET01930
SET01940
SET01950
SET01960
SET01970
SET01980
SET01990
SET02000
SET02010
SET02020
SET02030
SET02040
SET02050
SET02060
SET02070
SET02080
SET02090
SET02100
SET02110
SET02120
SET02130
SET02140
SET02150
SET02160
SET02170
SET02180
SET02190
SET02200
SET02210
SET02220
SET02230
SET02240
SET02250
SET02260
SET02270
SET02280
SET02290
SET02300
SET02310
SET02320
SET02330
SET02340
SET02350
SET02360
SET02370

FILE: SET19

```

C      MED1
C
400  M = NKTCHR(CARD,COL)
      READ(RRUNIT,380) MED1
      REWIND RRUNIT
      GO TO 105
C
C      MED2
C
410  M = NKTCHR(CARD,COL)
      READ(RRUNIT,380) MED2
      REWIND RRUNIT
      GO TO 105
C
C      CONVERT CARD
C
500  M=NKTCHR(CARD,COL)
      IF(M.EQ.BLNK) GO TO 105
      IF(M.EQ.FF) GO TO 510
530  WRITE(NPRT,520)
520  FORMAT(' ERROR ON CONVERT CARD')
      GO TO 105
510  J=FNID12(CARD,COL,EQUOM)

      M=NUMBER(CARD,COL,GTNOF,ZERO)
      COL=COL-1
      GO TO 500
C
C      *END*
C
420  CONTINUE
      DO 430 I=1,256
      IF(TRANS1(I).EQ.QQ) GO TO 450
430  CONTINUE
      GO TO 440
450  WRITE(NPRT,153)
      WRITE(NPRT,451)
451  FORMAT(//,10X,'DEFAULT TRANSFORMATION USED')
      CALL GTTRANS
C
440  CONTINUE
C
      IF(GOOF.EQ.1) GO TO 333
      DO 460 I=1,11
      DO 460 J=1,19
      IF(TY(I,J).EQ.-100) GOOF=1
      IF(GOOF.EQ.1) GO TO 333
460  CONTINUE
461  CONTINUE
C
      WRITE(NPRT,1000)
      WRITE(NPRT,1100) GTROU,GTROF
      WRITE(NPRT,1200) GTWRU,GTWRF
      WRITE(NPRT,1300) GTNOF
      IF (MSKKEY .EQ. 1) WRITE(NPRT,1030)
      IF (MSKKEY .EQ. 2) WRITE(NPRT,1031)
      IF (MSKKEY .EQ. 3) WRITE(NPRT,1032)
1000  FORMAT(//, 'USER HAS REQUESTED THE FOLLOWING OPTIONS :')
1030  FORMAT(' TRANSITION YEAR MASK')
1031  FORMAT(' PHASE THREE MASK')
1032  FORMAT(' INPUT MASK')
1100  FORMAT(' READ UNIT = ',I3,' READ FILE = ',I3)
1200  FORMAT(' WRITE UNIT = ',I3,' WRITE FILE = ',I3)
1300  FORMAT(' THE NUMBER OF TAPE FILES TO BE DUMPED TO DOT FILES = ',
      * I4)
C
      WRITE(NPRT,630)
630  FORMAT(//, ' THE CROP CODE TO SYMBOL TRANSFORMATION')
      FIRST=1
      SYM=TRANS1(1)
      DO 631 I=1,256
      LAST=I-1
      IF(TRANS1(I).NE.SYM) WRITE(NPRT,632) SYM,FIRST, LAST
      IF(TRANS1(I).NE.SYM) FIRST=I
      IF(TRANS1(I).NE.SYM) SYM=TRANS1(I)
      IF(I.EQ.256) WRITE(NPRT,632) SYM,FIRST,I
631  CONTINUE
632  FORMAT(1H,5X,A1,' = ',I3,' , ',I3)
      WRITE(NPRT,640)

```

SET02380
 SET02390
 SET02400
 SET02410
 SET02420
 SET02430
 SET02440
 SET02450
 SET02460
 SET02470
 SET02480
 SET02490
 SET02500
 SET02510
 SET02520
 SET02530
 SET02540
 SET02550
 SET02560
 SET02570
 SET02580
 SET02590
 SET02600
 SET02610
 SET02620
 SET02630
 SET02640
 SET02650
 SET02660
 SET02670
 SET02680
 SET02690
 SET02700
 SET02710
 SET02720
 SET02730
 SET02740
 SET02750
 SET02760
 SET02770
 SET02780
 SET02790
 SET02800
 SET02810
 SET02820
 SET02830
 SET02840
 SET02850
 SET02860
 SET02870
 SET02880
 SET02890
 SET02900
 SET02910
 SET02920
 SET02930
 SET02940
 SET02950
 SET02960
 SET02970
 SET02980
 SET02990
 SET03000
 SET03010
 SET03020
 SET03030
 SET03040
 SET03050
 SET03060
 SET03070
 SET03080
 SET03090
 SET03100
 SET03110
 SET03120
 SET03130
 SET03140
 SET03150
 SET03160

ORIGINAL PAGE IS
OF POOR QUALITY

FILE: SET19

```
640  FORMAT(//, ' THE MASK')  
      DO 641 I=1,11  
      WRITE(NPWT,642) (TY(I,J),J=1,19)  
641  CONTINUE  
642  FORMAT(1H, '5X,19I5)  
      RETURN  
C  
      END
```

SET03170
SET03180
SET03190
SET03200
SET03210
SET03220
SET03230
SET03240

ORIGINAL PAGE 1
OF 100 QUALITY

22. GTTCN PROCESSOR

FILE: GTTCN

| | | |
|---|--------------------------------------|-----------|
| C | WRITTEN BY C W AHLERS | GTTCN0010 |
| C | | GTTCN0020 |
| C | GROUND TRUTH TAPE CONVERSION ROUTINE | GTTCN0030 |
| C | | GTTCN0040 |
| | SUBROUTINE GTTCN(AHWAY, TOP) | GTTCN0050 |
| | IMPLICIT INTEGER (A-Z) | GTTCN0060 |
| | DIMENSION AHWAY(1) | GTTCN0070 |
| | CALL SET17 | GTTCN0080 |
| | CALL TCN(AHWAY, TOP) | GTTCN0090 |
| | RETURN | GTTCN0100 |
| | END | GTTCN0110 |

FILE: GTCRPL

```

C      WRITTEN BY C. W. AHLERS
      SUBROUTINE GTCRPL(CROP,MT,NC)
      IMPLICIT INTEGER (A-Z)
      DIMENSION MT(K)
      COMMON /GTRF/NRDP,NPRT,PRTKEY,VLB(6),GTRDU,GTRDF
      *GTRDU,GTRDF,GTNOF
C 901  WRITE(NPRT,901)
      FORMAT(1H,10X,'GTCRPL')
      NC=0
      DO 10 I=1,6
      IF (VLB(I).LT.1) GO TO 10
      CC=MT(I)
      N=0
      DO 20 J=1,6
      IF (VLB(J).LT.1) GO TO 20
      IF (CC.EQ.MT(J)) N=N+1
20    CONTINUE
      IF (N.LE.NC) GO TO 10
      NC=NC+1
      CROP=CC
      IF (NC.GE.3) RETURN
10    CONTINUE
      RETURN
      END

```

GTC00010
 GTC00020
 GTC00030
 GTC00040
 GTC00050
 GTC00060
 GTC00070
 GTC00080
 GTC00090
 GTC00100
 GTC00110
 GTC00120
 GTC00130
 GTC00140
 GTC00150
 GTC00160
 GTC00170
 GTC00180
 GTC00190
 GTC00200
 GTC00210
 GTC00220
 GTC00230
 GTC00240

ORIGINAL PAGE IS
OF POOR QUALITY

FILE: GTUNPK

| | | |
|-----|---|----------|
| | SUBROUTINE GTUNPK(IDATA,N1,N2,OS,NRPS,LENGTH) | GTU00010 |
| | IMPLICIT INTEGER (A-Z) | GTU00020 |
| | DIMENSION IDATA(1) | GTU00030 |
| C | WRITE(3,900) | GTU00040 |
| C | WRITE(6,900) | GTU00050 |
| 900 | FORMAT(1H,10X,'GTUNPK') | GTU00060 |
| | DO 10 I=1,NRPS | GTU00070 |
| | DO 20 J=N1,N2 | GTU00080 |
| | JJ=J+OS | GTU00090 |
| | IF(JJ.GT.3040) WRITE(6,901) JJ | GTU00100 |
| | IF(JJ.GT.3040) STOP | GTU00110 |
| 901 | FORMAT(1H,10X,'JJ=',110) | GTU00120 |
| | II=IDATA(JJ) | GTU00130 |
| | IF(II.GT.128) IDATA(JJ)=IDATA(JJ)-128 | GTU00140 |
| | IF(II.LE.128) IDATA(JJ)=128+IDATA(JJ) | GTU00150 |
| 20 | CONTINUE | GTU00160 |
| | OS=OS+LENGTH | GTU00170 |
| 10 | CONTINUE | GTU00180 |
| | RETURN | GTU00190 |
| | END | GTU00200 |

FILE: LINLAB

```

SUBROUTINE LINLAB(/IDATA/,MAXREC)
IMPLICIT INTEGER (A-Z)
DIMENSION MT(4),IDATA(3040)
C 901 WRITE(3,901)
      FORMAT(1H,10X,'LINLAB')
      OS=72
      DO 30 PIX=1.196
      OS=OS+2
      M=0
      DO 40 LL=1.3
      DO 50 SS=1.2
      M=M+1
      SSS=OS+SS-2
      IF (SSS.GT.3040) WRITE(6,903) SSS
      IF (SSS.GT.3040) STOP
C 903 WRITE(3,903) SSS
      FORMAT(1H,10X,'SSS=',I10)
      IF (M.GT.4) WRITE(6,911) M
      IF (M.GT.4) STOP
      911 FORMAT(1H,10X,'M=',I5)
      MT(M)=IDATA(SSS)
      50 CONTINUE
      OS=OS+MAXREC
      40 CONTINUE
      CALL GTCRPL(CROP,MT,NCROP)
C PP=PIX*72
      PP=PIX
      IDATA(PP)=CROP
      OS=72+2*PIX
      30 CONTINUE
      DO 60 I=242.2960
      IDATA(I)=0
C 60 WRITE(6,902) (IDATA(K),K=1.196)
      902 FORMAT(1H,2514)
      RETURN
      END

```

```

L N00010
L N00020
L N00030
L N00040
L N00050
L N00060
L N00070
L N00080
L N00090
L N00100
L N00110
L N00120
L N00130
L N00140
L N00150
L N00160
L N00170
L N00180
L N00190
L N00200
L N00210
L N00220
L N00230
L N00240
L N00250
L N00260
L N00270
L N00280
L N00290
L N00300
L N00310
L N00320
L N00330
L N00340
L N00350
L N00360

```

~~22-5~~
509

FILE: SET17

```

C
140 M = NATCHW(CARD,COL)
IF (M.EQ.V) GO TO 155
IF (M.EQ.BLNK) GO TO 105
142 WRITE(MUNT,153)
143 FORMAT(' ERROR ON LABEL CARD')
GO TO 105
155 J = FIND12(CARD,COL,EDUCOM)
IF (J.NE.2) GO TO 152
NOLAB = NUMRFX(CARD,COL,VLR,NOLAB)
GO TO 105

C
C
C
READ TAPE

140 M = NATCHW(CARD,COL)
IF (M.EQ.BLNK) GO TO 105
IF (M.EQ.U) GO TO 190
IF (M.EQ.EF) GO TO 200
145 WRITE(MUNT,147)
147 FORMAT(' ERROR ON HEAD TAPE CARD')
GO TO 105
190 J = FIND12(CARD,COL,EDUCOM)
IF (J.NE.2) GO TO 145
M = NUMRFX(CARD,COL,ATHDU,ZERO)
COL = COL - 1
GO TO 140
200 J = FIND12(CARD,COL,EDUCOM)
IF (J.NE.2) GO TO 145
M = NUMRFX(CARD,COL,GTADF,ZERO)
COL = COL - 1
GO TO 140

C
C
C
WRITE TAPE

210 CONTINUE
214 M = NATCHW(CARD,COL)
IF (M.EQ.BLNK) GO TO 105
IF (M.EQ.U) GO TO 230
IF (M.EQ.EF) GO TO 240
215 WRITE(MUNT,220)
220 FORMAT(' ERROR ON WRITE TAPE CARD')
GO TO 105
230 J = FIND12(CARD,COL,EDUCOM)
IF (J.NE.2) GO TO 215
M = NUMRFX(CARD,COL,GTWNU,ZERO)
COL = COL - 1
GO TO 214
240 J = FIND12(CARD,COL,EDUCOM)
IF (J.NE.2) GO TO 215
M = NUMRFX(CARD,COL,GTWNE,ZERO)
COL = COL - 1
GO TO 214

C
C
C
OPTION CARD

330 M = NATCHW(CARD,COL)
IF (M.EQ.BLNK) GO TO 105
IF (M.EQ.U) GO TO 340
333 WRITE(MUNT,335)
335 FORMAT(' ERROR ON OPTION CARD')
GO TO 105
340 PRKEY = 1
GO TO 105

C
C
C
DATE CARD

370 M = NATCHW(CARD,COL)
IF (M.EQ.BLNK) GO TO 105
DEAD(MUNT,440)DATE
380 FORMAT(10X,1544)
DEAD(MUNT,440)
GO TO 105

C
C
C
COMMENT CARD

390 M = NATCHW(CARD,COL)
IF (M.EQ.BLNK) GO TO 105
DEAD(MUNT,440)COMMENT
DEAD(MUNT,440)

```

```

SET000400
SET000410
SET000420
SET000430
SET000440
SET000450
SET000460
SET000470
SET000480
SET000490
SET000900
SET000910
SET000920
SET000930
SET000940
SET000950
SET000960
SET000970
SET000980
SET000990
SET001000
SET001010
SET001020
SET001030
SET001040
SET001050
SET001060
SET001070
SET001080
SET001090
SET001100
SET001110
SET001120
SET001130
SET001140
SET001150
SET001160
SET001170
SET001180
SET001190
SET001200
SET001210
SET001220
SET001230
SET001240
SET001250
SET001260
SET001270
SET001280
SET001290
SET001300
SET001310
SET001320
SET001330
SET001340
SET001350
SET001360
SET001370
SET001380
SET001390
SET001400
SET001410
SET001420
SET001430
SET001440
SET001450
SET001460
SET001470
SET001480
SET001490
SET001500
SET001510
SET001520
SET001530
SET001540
SET001550
SET001560
SET001570
SET001580

```

FILE: SET17

ORIGINAL PAGE IS
OF POOR QUALITY

```

C      GO TO 105
C
C      HED1
C
400  M = NATCHH(CARD,COL)
      READ(RRUNIT,380) HED1
      REWIND RRUNIT
      GO TO 105
C
C      HED2
C
410  M = NATCHH(CARD,COL)
      READ(RRUNIT,380) HED2
      REWIND RRUNIT
      GO TO 105
C
C      CONVERT CARD
C
500  M=NATCHH(CARD,COL)
      IF(M.EQ.-LNK) GO TO 105
      IF(M.EQ.-FF) GO TO 510
530  WRITE(NPRT,520)
520  FORMAT(' ERROR ON CONVERT CARD')
      GO TO 105
510  J=FINDI2(CARD,COL,EQUOM)

      M=NUMBER(CARD,COL,GTNOF,ZERO)
      COL=COL-1
      GO TO 500
C
C      *END*
C
420  CONTINUE
      IF (NOLAB .NE. 0) GO TO 440
      DO 430 I=1,6
      VLR(I) = 1
430  CONTINUE
      NOLAB = 6
      GO TO 440
450  WRITE(NPRT,153)
      NOLAB=0
      GO TO 420
C
440  CONTINUE
      IF(NOLAB.LT.6) GO TO 450
C
C
      WRITE(NPRT,1000)
      WRITE(NPRT,1100) GTROU,GTROF
      WRITE(NPRT,1200) GTWRU,GTWRF
      WRITE(NPRT,1300) GTNOF
      WRITE(NPRT,1010) (VLR(I),I=1,6)
      IF (PRTKEY .EQ. 1) WRITE(NPRT,1030)
1000  FORMAT('/// USER HAS REQUESTED THE FOLLOWING OPTIONS ://')
1010  FORMAT(' THE LABEL VECTOR IS = ',6I3)
1030  FORMAT(' PRINT THE 204 DOT LABELS')
1100  FORMAT(' READ UNIT = ',I3,' READ FILE = ',I3)
1200  FORMAT(' WRITE UNIT = ',I3,' WRITE FILE = ',I3)
1300  FORMAT(' THE NUMBER OF FILES TO BE CONVERTED = ',I3)
C
      RETURN
C
      END

```

SET01590
 SET01600
 SET01610
 SET01620
 SET01630
 SET01640
 SET01650
 SET01660
 SET01670
 SET01680
 SET01690
 SET01700
 SET01710
 SET01720
 SET01730
 SET01740
 SET01750
 SET01760
 SET01770
 SET01780
 SET01790
 SET01800
 SET01810
 SET01820
 SET01830
 SET01840
 SET01850
 SET01860
 SET01870
 SET01880
 SET01890
 SET01900
 SET01910
 SET01920
 SET01930
 SET01940
 SET01950
 SET01960
 SET01970
 SET01980
 SET01990
 SET02000
 SET02010
 SET02020
 SET02030
 SET02040
 SET02050
 SET02060
 SET02070
 SET02080
 SET02090
 SET02100
 SET02110
 SET02120
 SET02130
 SET02140
 SET02150
 SET02160
 SET02170
 SET02180
 SET02190
 SET02200
 SET02210
 SET02220

FILE: TCN

```

C      WRITTEN BY C W AMLEPS
      SUBROUTINE TCN(ARRAY,TOP)
      IMPLICIT INTEGER (A-Z)
      DIMENSION ARRAY(1)
      LOGICAL*1 VRL(2400),IDL(800)
      DIMENSION IDATA1(3060)
      DIMENSION IPUF(765)
      DIMENSION FETVEC(30)
      DATA FETVEC/30*0/
      DIMENSION IDATA(3060)
      DIMENSION BLOCK(6)
      COMMON /TAPERD/ IUNIT,IFIRST,FSCAN,SAMEND,SAMINC,READY,NSCAN,
      * LINC,ID(200),DSL,LBUF(30),JREC(30),IBYTE(30),NRUFS,FILENO,LINEND
      * LININC,NSAMP,NOCHAN,IFORMT
      COMMON /RTAP/TCOUNT,FORMAT,UNIT,VARBL(600),IREMD
      EQUIVALENCE (VRL,VARBL)
      EQUIVALENCE (ID,IDL)
C      LOGICAL*1 DY(4),MN(4),YR(4),ST(4)
C      EQUIVALENCE (DAY,DY(1)),(MON,MN(1)),(YEAR,YR(1)),(SITE,ST(1))
C      EQUIVALENCE (DAY,DY),(MON,MN),(YEAR,YR),(SITE,ST)
      COMMON /GTRK/NKOR,NPRT,PRTKEY,VLB(6),GTRDU,GTRDF,
      * GTWRU,GTWRF,GTNOF
      EQUIVALENCE (IDATA(1),ARRAY(1))
C      EQUIVALENCE (IDATA(1),ARRAY(3061))
      UNIT=GTWRU
      IUNIT=GTRDU
      FORMAT=IFORMT
      NEOF=0
      FILE=0
      NOCHAN=1
      NOFEAT=NOCHAN
      FETVEC(1)=1
      ZERO=0
      OUTPX=196
      INPX=392
      OSG=72
      ONE=1
      GTNRPO=3
      BLOCK(1)=1
      BLOCK(2)=351
      BLOCK(3)=1
      BLOCK(4)=1
      BLOCK(5)=392
      BLOCK(6)=1
10    CONTINUE
      PEWINO GTRDU
      PEWINO GTWRU
      RDF=GTRDF-1
      WRF=GTWRF-1
      CALL FSEFEL(GTWRU,WRF,ISTATW)
C 10    CONTINUE
      IPFC=0
      ICOUNT=0
      NPFC=0
      FILE=FILE+1
      DAY=0
      MON=0
      YEAR=0
      SITE=0
      CALL TAPHDR(GTRDU,RDF)
      DAY=ID(25)
      MON=ID(26)
      YEAR=ID(27)
      SITE=ID(28)
      VRL(73)=IDL(100)
      VRL(74)=IDL(104)
      VRL(75)=IDL(108)
      VRL(76)=IDL(111)
      VRL(80)=IDL(112)
      CALL WTHFD(NOFEAT,FETVEC,OUTPX,FORMAT,GTWRU)
      GTREC=540
      WRITE(NPRT,601) GTRDF,GTWRF
601    FORMAT(//,' FILE ',I4.5X,' BEING CONVERTED TO FILE ',I4)
      WRITE(NPRT,306) SITE,DAY,MON,YEAR
306    FORMAT(' SITE = ',I5.5X,' DAY=',I5.5X,' MONTH=',I5.5X,
      * ' YEAR=',I5)
      IF(PRTKEY.EQ.1) WRITE(NPRT,600)
      CALL FLDINT(BLOCK,FETVEC,NOFEAT)

```

TCN00010
 TCN00020
 TCN00030
 TCN00040
 TCN00050
 TCN00060
 TCN00070
 TCN00080
 TCN00090
 TCN00100
 TCN00110
 TCN00120
 TCN00130
 TCN00140
 TCN00150
 TCN00160
 TCN00170
 TCN00180
 TCN00190
 TCN00200
 TCN00210
 TCN00220
 TCN00230
 TCN00240
 TCN00250
 TCN00260
 TCN00270
 TCN00280
 TCN00290
 TCN00300
 TCN00310
 TCN00320
 TCN00330
 TCN00340
 TCN00350
 TCN00360
 TCN00370
 TCN00380
 TCN00390
 TCN00400
 TCN00410
 TCN00420
 TCN00430
 TCN00440
 TCN00450
 TCN00460
 TCN00470
 TCN00480
 TCN00490
 TCN00500
 TCN00510
 TCN00520
 TCN00530
 TCN00540
 TCN00550
 TCN00560
 TCN00570
 TCN00580
 TCN00590
 TCN00600
 TCN00610
 TCN00620
 TCN00630
 TCN00640
 TCN00650
 TCN00660
 TCN00670
 TCN00680
 TCN00690
 TCN00700
 TCN00710
 TCN00720
 TCN00730
 TCN00740
 TCN00750
 TCN00760
 TCN00770
 TCN00780
 TCN00790

ORIGINAL PAGE IS
OF POOR QUALITY

FILE: TCN

| | | |
|-----|--|----------|
| 600 | FORMAT(' THE 209 DOT LABELS') | TCN00800 |
| 20 | CONTINUE | TCN00810 |
| | DO A01 J=1,3 | TCN00820 |
| | IREC=IREC+1 | TCN00830 |
| | IF(IREC.GT.200) GO TO 50 | TCN00840 |
| | CALL LINERD(IDATA1,ENDTAP) | TCN00850 |
| C | IF(ENDTAP.EQ.-1) GO TO 50 | TCN00860 |
| | DO A00 I=1,GTREC | TCN00870 |
| | JJ=I+GTREC*(J-1)+72 | TCN00880 |
| | IDATA(JJ)=IDATA1(I) | TCN00890 |
| 800 | CONTINUE | TCN00900 |
| 801 | CONTINUE | TCN00910 |
| | NFOF=0 | TCN00920 |
| | NREC=NREC+1 | TCN00930 |
| | OSGG=OSG | TCN00940 |
| | CALL GTUNPK(IDATA,ONE,OUTPX,OSGG,GTNRPD,GTREC) | TCN00950 |
| | G3=GTREC*3 | TCN00960 |
| C | IF(NREC.LE.2) WRITE(NPRT,850) (IDATA(K),K=1,63) | TCN00970 |
| 850 | FORMAT(1H,15I4) | TCN00980 |
| | CALL LINLAR(IDATA,GTREC) | TCN00990 |
| | N10=MOD(NREC,10) | TCN01000 |
| C | IF(NREC.EQ.1) WRITE(NPRT,850) (IDATA(K),K=1,GTREC) | TCN01010 |
| | CALL WRTLN(IDATA,ZERO) | TCN01020 |
| | IF(PRTKEY.EQ.1.AND.N10.EQ.0) WRITE(NPRT,839) (IDATA(I),I=10,190) | TCN01030 |
| | 10 | TCN01040 |
| 839 | FORMAT(1H,5X,19I5) | TCN01050 |
| | GO TO 20 | TCN01060 |
| 50 | CONTINUE | TCN01070 |
| | ENDFILE GTWRU | TCN01080 |
| | WRITE(NPRT,700) NREC | TCN01090 |
| 700 | FORMAT(' NUMBER OF SCAN LINES WRITTEN=',I5) | TCN01100 |
| | WRITE(NPRT,750) GTRDF,GTWRF | TCN01110 |
| 750 | FORMAT(' FILE ',I5,5X,' WAS CONVERTED ONTO FILE ',I5) | TCN01120 |
| | GTRDF=GTRDF+1 | TCN01130 |
| | RDF=RDF+1 | TCN01140 |
| | GTWRF=GTWRF+1 | TCN01150 |
| | IF(FILE.LT.GTNOF) GO TO 10 | TCN01160 |
| | ENDFILE GTWRU | TCN01170 |
| | REWIND GTWRU | TCN01180 |
| | REWIND GTWRU | TCN01190 |
| | WRITE(NPRT,250) | TCN01200 |
| 250 | FORMAT(' PROGRAM GTTCN RUN COMPLETED') | TCN01210 |
| | RETURN | TCN01220 |
| | END | TCN01230 |

ORIGINAL PAGE IS
OF POOR QUALITY

23. TESTSP PROCESSOR

FILE TESTSP

```

SUBROUTINE TESTSP (ARRAY, TOP)
C*
C* .....
C* THIS PROGRAM PERFORMS A MODIFIED VERSION OF THE CLUSTERING
C* ALGORITHM (ISODATA) ORIGINALLY DEVELOPED BY BALL AND HALL OF
C* STANFORD RESEARCH INSTITUTE. THE ALGORITHM HAS BEEN MODIFIED
C* ON THE RECOMMENDATIONS OF ED KAN (LEC).
C*
C* THE PROGRAM EXPECTS MULTISPECTRAL SCANNER DATA
C* IN EITHER THE LAPSYS 22 OR THE UNIVERSAL
C* FORMAT. THE DATA TAPE SHOULD BE ASSIGNED TO FORTRAN UNIT 3.
C* .....
C*
C* IMPLICIT INTEGER (A-X)
C* INCLUDE COMBKS.LIST
C* INCLUDE COMNTS.LIST
C* INCLUDE COMMK6.LIST
C* INCLUDE CMRK16.LIST
C* COMMON/PASS/STOP, LNCAT, NMIN, KRN, STD MAX, DL MIN, SEP,
C*   MAP, SPTRIG, IRD, KPTS, NOPTS, PUNCH,
C*   ICHN, CHNTHS, ICHAIN(62), NWDS, IREGIN, BEGIN1,
C*   BEGIN2, BEGIN3, CLSNAM, NOFLD, IPT, TOTWRD, TOTPTS,
C*   NCLASS, NOCLS, TOTSUR, TOTFLD, TOTVRT, NOCL, NVRT,
C*   NXTCLS, NOFEAT, MAXCLS, FFTVEC(30), SYMMTX(62),
C*   VARSIZ, STATKY, ISOKEY, MAPFMT, MAPKEY, SEQUEN(20), PERCEN, SIMERP,
C*   IORDN, INUNIT, INFILE, INITM, PMIN, SUBVEC(62), NOSUB2, CHNVC(30),
C*   NOCHAN, ERCOMP, NOSEQ, MEANDU, MEANDU,
C*   SYMDO, SYMDU, ITRIGO, ITRIGU, DOFLAG,
C*   DUFLAG, DODU, STDOTS(60), NSDOTS, SUNCOR(30), LLNCAT,
C*   OVERT(250, 2), DRECT(160, 2), DVPNT(11, 2), IDCNT(2), NDOU(2),
C*   MXFET1, MAXPOP
C* REAL SUNCOR
C*
C* COMMON BLOCK 'PASS' IS USED ONLY BY THE ISOCLS PROCESSOR.
C*
C* ISOCLS USES THE RANDOM ACCESS DRUM FILE AS FOUR DISTINCT FILES.
C* SEE DEFINITIONS OF IREGIN, BEGIN1, BEGIN2, BEGIN3 BELOW
C*
C* DEFINITIONS
C*
C*   ISTOP - MAX. NO. OF ITERATIONS FOR THE CLUSTERING PROCEDURE
C*           SET IN SETUP7 ROUTINE. (USER INPUT)
C*   LNCAT - CURRENT NO. OF CLUSTERS. SET INITIALLY IN RDFILE OR ISTES
C*           ISOCLS. THEN ONLY IN ISODAT.
C*   NMIN - MIN. NO. OF POINTS TO ALLOW PER CLUSTER
C*           SET IN SETUP7 ROUTINE. (USER INPUT)
C*   KRN - PRINT CLUSTER SUMMARY EVERY 'KRN' ITERATION(S)
C*           SET IN SETUP7 ROUTINE. (USER INPUT)
C*   STD MAX - STANDARD DEVIATION FOR SPLITTING CLUSTERS
C*           SET IN SETUP7 ROUTINE. (USER INPUT)
C*   DL MIN - MIN. DISTANCE BETWEEN CLUSTERS FOR COMBINING.
C*   SEP - DISTANCE TO SEPARATE CLUSTERS. SET EITHER IN SETUP7,
C*           BY USER INPUT, OR IN ID
C*           BY USER INPUT, OR IN ISODAT.
C*   MAP - PRINT A CLUSTER MAP EVERY 'MAP' ITERATION(S) - SETUP7
C*   SPTRIG - TRIGGER TELLING WHETHER OR NOT 'SEP' WAS INPUT. -SETUP7
C*   IRD - NO. OF RECORDS TO READ FROM DATA FILE. COMPUTED IN
C*           ISOCLS
C*   NOPTS - NO. OF POINTS IN EACH RECORD. COMPUTER IN ISOCLS
C*   CONTINUE
C*   KPTS - NO. OF POINTS IN LAST RECORD. COMPUTER IN ISOCLS
C*   PUNCH - TRIGGER TELLING WHETHER OR NOT TO PUNCH THE MODULE
C*           STAT DECK. - SETUP7
C*   ICHN - TRIGGER TELLING WHETHER OR NOT CHAINING IS TO BE DONE
C*   CHNTHS - MIN. DISTANCE BETWEEN CLUSTERS FOR CHAINING - SETUP7
C*   ICHAIN - ARRAY CONTAINING CHAINED CLUSTER NUMBERS. SET IN
C*           'ICHAIN' ROUTINE.
C*   NWDS - TOTAL NO. OF WORDS AVAILABLE FOR DRUM STORAGE OF
C*           IMAGE DATA TO BE CLUSTERED - SET IN ISOCLS
C*   IREGIN - BEGINNING DRUM FILE ADDRESS FOR INPUT INITIAL CLUSTER
C*           CENTERS - SET IN ISOCLS
C*   REGINS - BEGINNING DRUM FILE ADDRESS FOR TEMPORARY STORAGE OF
C*           CLASS STATISTICS - SET IN ISOCLS ROUTINE
C*   BEGIN1 - BEGINNING DRUM FILE ADDRESS FOR IMAGE DATA

```

FILE TESTSP

```

C*      BEGIN2 - BEGINNING DRUM FILE ADDRESS FOR 'IPLACE' .(CLUSTER TO TES00770
C*      WHICH CORRESPONDING POINT BELONGS.) TES00780
C*      CLSNAM - NAME OF CLASS CURRENTLY BEING PROCESSED. - RDDATA TES00790
C*      NOFLD - NO. OF FIELDS INPUT FOR THIS CLASS - RDDATA TES00800
C*      IPT - NO. OF WORDS OF STORAGE USED IN 'ARRAY' FOR FIELD AND TES00810
C*      CLASS INFORMATION FOR THIS CLASS. - RDDATA TES00820
C*      TOTWRD - TOTAL WORDS WRITTEN ON DRUM FILE BEGINNING AT ADDRESS TES00830
C*      REGIN1 - RDDATA TES00840
C*      TOTPTS - TOTAL POINTS TO BE CLUSTERED FOR CURRENT CLASS - RDDATA TES00850
C*      NCLASS - NO. OF CLASSES TO BE CLUSTERED FOR CURRENT CALL TO TES00860
C*      ISOCLS - USER INPUT - SETUP7. TES00870
C*      NOCLS - CURRENT CLASS NO. - ISOCLS TES00880
C*      TOTSUB - TOTAL CLUSTERS FOR THIS CALL TO ISOCLS TES00890
C*      TOTFLD - TOTAL FIELDS FOR ALL CLASSES - ISOCLS TES00900
C*      TOTVRT - TOTAL VERTICES FOR ALL FIELDS - ISOCLS TES00910
C*      NOCL - NO. OF CLASSES SINCE LAST CALL TO SETUP - RDDATA TES00920
IEY0331 COMMENTS DELETED *****
COMMON/GLOBAL/HEAD(63),MAPTAP,DATAPE,SAVTAP,BMFILE,BMKEY, TES01200
* HISFIL,HISKEY,TRFORM,ERIP,ERPKEY,MAPUNT,NOFILE, TES01210
* DRUMAD,DRUMDS,PAGSIZ,DATFIL,STAFIL,ASAV,ASAVFL TES01220
* ,NHSTUN,NHSTFI,SCTRUN,MAPFIL TES01230
* ,DOTUNT,DOTFIL,NCHPAS,TRNSFL,BMTRFL,HISTFL,PCHUNT, TES01240
* CRDUNT,PRUNT,RANDIO TES01250
COMMON/ISOLNK/SUNANG(8),ISUNT,ISUNC,SMSTR,SMSTP,SMINC,LINSKP TES01260
CSEND TES01270
DIMENSION KVAR(11500) TES01280
KVARDM = 11500 TES01290
DIMENSION ARRAY(1) TES01300
DIMENSION COVAR(465) TES01310
DIMENSION NN(60) TES01320
DATA SYMDA /* /*,SYMDB /* /* /* TES01330
MAXPOP=62 TES01340
MXFET1=30 TES01350
IBEGIN=DRUMAD TES01360
C* TES01370
C* RESERVE ENOUGH DRUM STORAGE FOR MAXIMUM INITIAL MEANS TES01380
C* TES01390
C* BEGIN3=IBEGIN + MAXPOP*MXFET1 + MXFET1 + 2 TES01400
C* TES01410
C* CALL SETUP TO READ CARD INPUT AND INITIALIZE DEFAULT VALUES TES01420
C* TES01430
C* TES01440
ITIME=1 TES01450
NOCLS = 0 TES01460
TOTFLD = 0 TES01470
TOTVRT = 0 TES01480
TOTSUB = 0 TES01490
CORBAS=1 TES01500
ITRIGU = 0 TES01510
ITPIGO=0 TES01520
SYMD0 = SYMDA TES01530
SYMDU = SYMDR TES01540
MEAND0 = 0 TES01550
MEANDU = 255 TES01560
1 CALL SETUP7(ARRAY(CORBAS),TOP,ITIME) TES01570
IDUM = MAXCLS TES01580
IF(ITIME.GT.1)GO TO 2 TES01590
VARSIZ=NOFEAT*(NOFEAT+1)/2 TES01600
REGIN1 = REGIN3 + NCLASS*MAXPOP*(VARSIZ + NOFEAT + 1) TES01610
NWDS=DRUMDS-(BEGIN1-DRUMAD) TES01620
2 ITIME=ITIME+1 TES01630
NOCL=0 TES01640
C* TES01650
C* CALL RDDATA TO COORDINATE READING OF DATA TES01660
C* TES01670
5 MAXDIM = TOP-CORBAS TES01680
FD1=CORBAS TES01690
CALL RDDPAT(FD1,MAXDIM,KVAR,KVARDM,LAST) TES01700
MAXCLS = IDUM + DODU TES01710
WRITE(4,210) NDOU(1),NDOU(2) TES01720
210 FORMAT(1X,/' DO/DO CLUSTER POP FOR THIS CLASS ',217) TES01730
REGIN2 = BEGIN1 + (TOTWRD/4) + 2 TES01740
N1 = FD1 + IPT TES01750
MEANS1=N1 + MAXCLS TES01760
STDEV1=MEANS1 + MAXCLS*NOFEAT TES01770
ITOP = STDEV1 + MAXCLS*NOFEAT TES01780
MAXDIM=TOP-ITOP

```

FILE TESTSP

```

      NOPTS = MAXDIM/(NOFEAT+1)
      NOPTS = (NOPTS/4)*4
C     PART OF PACKING CHG SEPT 1978
      IDAT1 = TTOP
      IF (NSDOTS.EQ.0) GO TO 4
      DOTDMF = NOCHAN
      TYPST = 1
      CALL RDDOTS(ARRAY(MEANS1),STDOTS,NSDOTS,
      * TYPST,DOTDMF,DOTDMC,DOTDUM,COVAR,
      * NOCHAN,CHNVC,UTDM,COVAR,
      * DOTDM,DOTDM,DOTDM,DOTDM,DOTDM,DOTDM,KVAR)
      LNCAT = NSDOTS
      DO 500 I = 1,NSDOTS
      DO 500 K = 1,NOFEAT
      III = (I-1)*NOFEAT + K
      II = III + MEANS1 - 1
500   ARRAY(II) = KVAR(III)
      IF (NOCHAN.EQ.NOFEAT) GO TO 8
      WRITE(6,110)
110   FORMAT(1H,'NO CHANNELS FOR STARTING NOT EQUAL THAT FOR CLUSTER')
      GO TO 9
4     CONTINUE
      IF (ISOKEY.EQ.1)GO TO 7
C*   SURVEC-SURCLASSES FROM STATISTICS FILE FOR INTIAL MEANS.
C*   NOSUB2=NUMBER OF INITIAL MEANS.
C*   CHNVEC=NUMBER OF CHANNELS FROM STATISTICS FILE. NOCHAN MUST EQUAL
      IF (INITM.EQ.1)GO TO 6
      LNCAT=1
      GO TO 8
6     LNCAT=NOSUB2
      CALL GETST(INUNIT,INFILE,ARRAY(MEANS1),DUM,NOSUB2,SURVEC,NOCHAN
      * ,CHNVC,ARRAY(TTOP),COVAR,0)
      LNCAT = NOSUB2
      GO TO 8
7     CONTINUE
      IF (ISOKEY.EQ.1)CALL RDFILE(ARRAY(MEANS1),ARRAY(TTOP))
8     CONTINUE
      IF (NOPTS.GT.0)GOTO 10
      WRITE(6,100)MAXDIM
100   FORMAT(' DIMENSION LIMITS EXCEEDED IN ISOCLS BY',16,
      * ' REDUCE CHANNELS OR MAX.CLUSTERS')
9     CALL CMERR
10    CONTINUE
      IRD=TOTPTS/NOPTS
      IF (MOD(TOTPTS,NOPTS).EQ.0)GO TO 20
      KPTS=MOD(TOTPTS,NOPTS)
      IRD=IRD+1
      IF (IRD.EQ.1)NOPTS=KPTS
      GO TO 25
20    KPTS=NOPTS
25    CONTINUE
C*   CALL ISODAT TO PERFORM CLUSTERING
C*
C*   A1=1
C*   A2=A1+ MAXCLS*NOFEAT
C*   CLD1=A2 + MAXCLS*NOFEAT
C*   KPLCE = NOPTS*NOFEAT + IDAT1
C*   CALL ISOPAT(IDAT1,ARRAY(KPLCE),ARRAY(MEANS1),ARRAY(N1),
C*   *   ARRAY(STDEV1),KVAR (CLD1),ARRAY(FD1),KVAR(A1),
C*   *   KVAR(A2))
C*
C*   CHAIN CLUSTERS WHOSE DISTANCES ARE LESS THAN DLMIN
C*
C*   LNCAT=LNCAT+DONU
C*   IF (ICHN.GT.0)CALL CHAIN(KVAR(CLD1))
C*
C*   PRINT FINAL RESULTS
C*
C*   CALL PRINT(-1,ARRAY(KPLCE),ARRAY(MEANS1),ARRAY(STDEV1),
C*   *   KVAR (CLD1),ARRAY(FD1),ARRAY(N1))
C*
C*   CREATE MAP OUTPUT TAPE FOR PMIS DAS IF DESIRED
C*
C*   IF (MAPFMT.GT.0)CALI. DSTAPE(ARRAY(KPLCE),KVAR(1),ARRAY(MEANS1),
C*   *   ARRAY(FD1))

```

TES01790
 TES01800
 TES01810
 TES01820
 TES01830
 TES01840
 TES01850
 TES01860
 TES01870
 TES01880
 TES01890
 TES01900
 TES01910
 TES01920
 TES01930
 TES01940
 TES01950
 TES01960
 TES01970
 TES01980
 TES01990
 TES02000
 TES02010
 TES02020
 TES02030
 TES02040
 TES02050
 TES02060
 TES02070
 TES02080
 TES02090
 TES02100
 TES02110
 TES02120
 TES02130
 TES02140
 TES02150
 TES02160
 TES02170
 TES02180
 TES02190
 TES02200
 TES02210
 TES02220
 TES02230
 TES02240
 TES02250
 TES02260
 TES02270
 TES02280
 TES02290
 TES02300
 TES02310
 TES02320
 TES02330
 TES02340
 TES02350
 TES02360
 TES02370
 TES02380
 TES02390
 TES02400
 TES02410
 TES02420
 TES02430
 TES02440
 TES02450
 TES02460
 TES02470
 TES02480
 TES02490
 TES02500
 TES02510
 TES02520
 TES02530
 TES02540

FILE TESTSP

```

C*      LNCAT=LNCAT-DODU
C*      CALCULATE COVARIANCE MATRIX FOR EACH CLUSTER
C*      IF(VARSIZ*LNCAT.GT.KVARDM)GO TO 30
C*      CALL COVPAT(KVAR,1DAT1,ARRAY(KPLCE),ARRAY(MEANS1),
C*      ARRAY(N1),IBAD)
C      CHECK FOR A CLUSTER DELETED FOR SINGULAR MATRIX
C      IF(IBAD.NE.0)STOP=0
C      IF(IBAD.NE.0)GO TO 25
C*      DO 26 II=1,LNCAT
26      NN(TOTSUB+II) = ARRAY(N1+II-1)
      TOTSUB = TOTSUB + LNCAT
      NOCLS = NOCLS + 1
      TOTFLD = TOTFLD + NOFLD
      TOTVRT = TOTVRT + NVRT
      ARRAY(FD1+1)=IPT + FD1
      ARRAY(FD1+2)=LNCAT
      ARRAY(FD1+3)=NOFLD
C*      WRITE STATS FOR THESE CLUSTERS ON SCRATCH FILE 18
C*      IF(NOCLS.EQ.1) ADRES=REGIN3
C*      IN=NOFEAT*LNCAT
C*      CALL RWRITE(ADRES,ARRAY(MEANS1),IN,JSTAT)
C*      ADRES=ADRES+IN
C*      IN=VARSIZ*LNCAT
C*      CALL RWRITE(ADRES,KVAR,IN,LSTAT)
C*      ADRES=ADRES+IN
C*      WAIT FOR I/O COMPLETION
60      IF(LSTAT.EQ.1) GO TO 60
C*      GO READ IN ANOTHER CLASS
C*      CORRAS=CORRAS+IPT
C*      IF(LAST.NE.1)GO TO 5
C*      IF(NOCLS.LT.NCLASS)GO TO 1
C*      NOW READ SCRATCH FILE AND STORE ON SAVTAP FILE AND PUNCH ON
C*      CARDS IF REQUESTED.
C*      FLD1 = 1
C*      VERTX1 = FLD1 + TOTFLD*4
C*      CLSNM1 = VERTX1 + TOTVRT*2
C*      NOSUR1 = CLSNM1 + NOCLS
C*      SURNM1 = NOSUR1 + NOCLS
C      RETRIEVE INFORMATION FROM 'ARRAY'
C      CALL GETINF(ARRAY(1),KVAR(FLD1),KVAR(VERTX1),KVAR(CLSNM1),
C      KVAR(NOSUB1),KVAR(SUBNM1),NOCLS,TOTSUB)
C      SWITCH = 1
C      OUTPUT STATS
C      CALL LABMAN(SAVTAP,STAFIL,NOCLS,TOTSUB,NOFEAT,TOTFLD,TOTVRT,
C      * FETVEC,KVAR(FLD1),KVAR(VERTX1),KVAR(CLSNM1),KVAR(NOSUB1),
C      * KVAR(SUBNM1),NN,REGIN3,VARSIZ,PUNCH,DUMMY,STATKY,SWTCH)
C      RETURN
30      KV=KVARDM
      WRITE(6,200)KV
      CALL CMERR
200      FORMAT(' DIMENSION LIMIT OF ',I6,' FOR COVARIANCES EXCEEDED')
      RETURN
      END

```

TES02550
 TES02560
 TES02570
 TES02580
 TES02590
 TES02600
 TES02610
 TES02620
 TES02630
 TES02640
 TES02650
 TES02660
 TES02670
 TES02680
 TES02690
 TES02700
 TES02710
 TES02720
 TES02730
 TES02740
 TES02750
 TES02760
 TES02770
 TES02780
 TES02790
 TES02800
 TES02810
 TES02820
 TES02830
 TES02840
 TES02850
 TES02860
 TES02870
 TES02880
 TES02890
 TES02900
 TES02910
 TES02920
 TES02930
 TES02940
 TES02950
 TES02960
 TES02970
 TES02980
 TES02990
 TES03000
 TES03010
 TES03020
 TES03030
 TES03040
 TES03050
 TES03060
 TES03070
 TES03080
 TES03090
 TES03100
 TES03110
 TES03120
 TES03130
 TES03140
 TES03150
 TES03160
 TES03170
 TES03180
 TES03190
 TES03200
 TES03210
 TES03220
 TES03230

FILE COVPAT

```

      IPACK1 = IPACK1*IPACK
      DUM = IPACK1
      KK=KK+1
      COVAR(KK,ICLS)=COVAR(KK,ICLS)+DUM
40  CONTINUE
45  CONTINUE
      IRC=IRC-1
      IF (IRC.GT.0) GO TO 20
      DO 50 I=1,LNCAT
      IF (N(I).EQ.0) GO TO 50
      KK=0
      DO 50 J=1,NOFEAT
      DO 50 K=1,J
      KK=KK+1
      COVAR(KK,I)=COVAR(KK,I)/N(I) - MEANS(K,I)*MEANS(J,I)
50  CONTINUE
      IACEPT=PMIN+NOFEAT
      IF (IACEPT.LT.NOFEAT) GO TO 58
C
C
      CHECK FOR SINGULAR COVARIANCE MATRIX
      DO 51 I=1,LNCAT
      CALL CHLDET(COVAR(I,I),NOFEAT,DUMM,DET)
      IF (DET.LT.TOL) GO TO 52
51  CONTINUE
      GO TO 58
C
C
      DELETE SINGULAR COVARIANCE CLUSTER
52  WRITE(6,160)I
      IF (LNCAT.EQ.1) CALL CMERR
      IRAD=1
      LNCAT=LNCAT-1
      LLNCAT=LLNCAT-1
      DO 53 II=1,LNCAT
      DO 53 III=1,NOFEAT
      MEANS(III,II)=MEANS(III,II+1)
53  CONTINUE
58  RETURN
160  FORMAT(2X,'CLUSTER',I5,' DELETED FOR SINGULARITY')
C
      IF (STATKY.NE.1) RETURN
55  WRITE(6,HEAD)
      WRITE(6,150)CLSNAM
      DO 60 I=1,LNCAT
      WRITE(6,90)I
      DO 70 LOC=1,NOFEAT,12
      ISTOP=LOC+11
      IF (ISTOP.GT.NOFEAT) ISTOP=NOFEAT
      WRITE(6,140) (CH,FETVEC(J),J=LOC,ISTOP)
      II=1
      KINC=1
      DO 60 J=LOC,NOFEAT
      K=J*(J+1)/2-II+1
      JK=K+KINC-1
      WRITE(6,100) (COVAR(M,I),M=K,JK)
      II=II+1
      KINC=KINC+1
60  IF (KINC.LT.ISTOP.AND.KINC.LT.12) KINC=KINC+1
      WRITE(6,110)
70  CONTINUE
80  CONTINUE
      RETURN
90  FORMAT(//,' COVARIANCE MATRIX FOR CLUSTER',I4/)
100  FORMAT(/6X,12F9.2)
110  FORMAT(//)
120  FORMAT(1H1)
140  FORMAT(9X,12(A3,I2,' '),3X))
150  FORMAT(/,' COVARIANCES FOR CLASS',2X,A4//)
      END

```

COV00770
 COV00780
 COV00790
 COV00800
 COV00810
 COV00820
 COV00830
 COV00840
 COV00850
 COV00860
 COV00870
 COV00880
 COV00890
 COV00900
 COV00910
 COV00920
 COV00930
 COV00940
 COV00950
 COV00960
 COV00970
 COV00980
 COV00990
 COV01000
 COV01010
 COV01020
 COV01030
 COV01040
 COV01050
 COV01060
 COV01070
 COV01080
 COV01090
 COV01100
 COV01110
 COV01120
 COV01130
 COV01140
 COV01150
 COV01160
 COV01170
 COV01180
 COV01190
 COV01200
 COV01210
 COV01220
 COV01230
 COV01240
 COV01250
 COV01260
 COV01270
 COV01280
 COV01290
 COV01300
 COV01310
 COV01320
 COV01330
 COV01340
 COV01350
 COV01360
 COV01370
 COV01380
 COV01390
 COV01400
 COV01410
 COV01420
 COV01430
 COV01440

```

SUBROUTINE ISOPAT (IDAT1, IPLACE, MEANS, N, STDEV, CLD, FLDINF, AVP, AMN)
IMPLICIT INTEGER (A-Z)
IMPLICIT INTEGER (A-Z)
INCLUDE COMBK5.LIST
INCLUDE COMBK6.LIST

COMMON/PASS/STOP, LNCAT, NMIN, KRN, STDMAX, DLMIN, SEP,
* MAP, SPTRIG, IRD, KPTS, NOPTS, PUNCH,
* ICHN, CHNTHS, ICHAIN(62), NWDS, IBEGIN, BEGIN1,
* REGIN2, REGIN3, CLSNAM, NOFLD, IPT, TOTWTD, TOTPTS,
* NCLASS, NOCLS, TOTSUH, TOTFLD, TOTVRT, NOCL, NVRT
* ,NXTCLS, NOFEAT, MAXCLS, FETVEC(30), SYMMTX(62)
* ,VARS17, STATKY, ISOKEY, MAPFMT, MAPKEY, SEQUEIN(20), PERCEN, SIMERP
* ,IORDER, ININIT, INFILE, INITM, PMIN, SUBVEC(62), NOSUB2, CHNVC(30)
* ,NOCHAN, FRCOMP, NOSEQ, MEANDU, MEANDU,
* SYMDU, SYMDU, ITRIGO, ITRIGU, DOFLAG,
* DUFLAG, DODU, STDOTS(60), NSDOTS, SUNCOR(30), LLNCAT,
* DVFT(250,2), DPECT(60,2), DVPNT(11,2), IDCNT(2), NDOU(2)
* ,MXFFT1, MAXPOP
REAL SUNCOR
COMMON/GLOBAL/HEAD(63), MAPTAP, DATAPE, SAVTAP, BMFILE, BMKEY,
* HISFIL, HISKEY, INFORM, ERIP, ERPKEY, MAPUNT, NOFILE,
* DPHMAN, DRMADS, PAGES17, DATFIL, STAFIL, ASAV, ASAVL
* ,NHSTIN, NHSTFI, SCTRUN, MAPFIL
* ,DOTUNT, DOTFIL, NCHPAS, TRNSFL, BMTRFL, HISTFL, PCHUNT,
* CRDUNT, PRUNT, RANDIO
COMMON/ISOLNK/SUNANG(8), ISUNT, ISUNC, SMSTR, SMSTP, SMINC, LINSKP

EQUIVALENCE (SGMIN, STDMAX)
REAL MEANS, STDEV, STDMAX, SEP, AVP, AMN, SGMA, RND,
* TEST, DMIN, DLMIN, CLD, TIME, PERCEN, DIJ
REAL ESUM, ESQT, MEAN(30,62), SDIJ
LOGICAL DEL
DIMENSION AVP(NOFEAT, MAXCLS), ISGMA(62)
DIMENSION AMN(NOFEAT, MAXCLS), SGMA(62)
DIMENSION IPLACE(NOPTS)
DIMENSION MEANS(NOFEAT, MAXCLS), N(MAXCLS)
DIMENSION STDEV(NOFEAT, MAXCLS), CLD(MAXCLS, MAXCLS)
DIMENSION FLDINF(1)
REAL SDUM
DIMENSION PTR(62)
DATA SS/'S'/'CC'/'C'/'
EQUIVALENCE (KDIM, NOFEAT), (LNCAT, INCAT)
DEL=.FALSE.
ISEQ=0
MAXCL = MAXCLS - DODU
IDUM=LNCAT+DODU-MAXCLS
IF (IDUM.GT.0) LNCAT=LNCAT-IDUM
ISTOP=STOP
SPLFIN=0
KKT=1
DO 5 I = 1,30
SUNCOR(I) = 1.
IF (ISUNC.NE.0.OR.ISUNT.NE.0) CALL SUNFAC(SUNCOR,SUNANG,
* FETVEC,NOFEAT,ISUNC,ISUNT.)

LK=K
ASSIGN DATA TO CLUSTERS

10 CONTINUE
LLNCAT = LNCAT + DODU
IF (DOFLAG.EQ.0) GO TO 12
DO 11 J=1,NOFEAT
MEANS(J,LNCAT+1)=MEANDU
11 IF (DUFLAG.EQ.0) GO TO 14
DO 13 J=1,NOFEAT
MEANS(J,LLNCAT) = MEANDU
13 CONTINUE
DO 15 K=1,LLNCAT
DO 15 J=1,NOFEAT
15 MEAN(J,K) = MEANS(J,K)
IF (LNCAT.LE.1.AND.KKT.GT.1) GO TO 530
CALL PSPPAT(MEANS,STDEV,N,CLD,IDAT1,IPLACE,AVP,AMN,MEANS)
CALL CLOCK(TIME)
IF (MOD(KKT,KRN).EQ.0) WRITE(6,120)KKT,TIME
120 FORMAT(' CUMULATIVE TIME AFTER ASSIGNING DATA TO CLUSTERS FOR ITER
* ATION',14,' IS',F10.6)

```

FILE: ISOPAT

```

      IF (ERCOMP.NE.1) GO TO 135
      ESUM=0.0
      DO 132 J=1,NOFEAT
      DO 133 K=1,LNCAT
      ESUM=ESUM+N(K)*(STDEV(J,K))**2/TOTPTS
132  CONTINUE
      ESOT=ESOT+(ESUM/NOFEAT)
      WRITE(6,133) ESOT,PERCEN,STOMAX
133  FORMAT(1X,///' ERCOMP= ',F7.3,' PERCEN = ',F5.3,' STOMAX = ',
      *F7.3/)
C*
C* CALCULATE DISTANCES BETWEEN CLUSTER CENTERS
C*
135  CALL CLDIST(CLD,STDEV,MEANS)
C*
C* IF STOP EQUALS ZERO DELETE SMALL CLUSTERS
C*
      LNCAT=LLNCAT
      IF (MOD(KKT,MAP)) 150,140,150
140  CALL PRINT(KKT,IPLACE,MEANS,STDEV,CLD,FLDINF,N)
      GO TO 161
150  IF (MOD(KKT,KRN)) 161,160,161
160  CONTINUE
      CALL PRINT(KKT,IPLACE,MEANS,STDEV,CLD,FLDINF,N)
161  CONTINUE
      LNCAT=LLNCAT-DODU
      IF (STOP.EQ.0) GO TO 162
C
C FOR ITERATION N CHECK N(K) AGAINST PMIN + NOFEAT
C
      IF (ISEQ.NE.NOSEQ) GO TO 169
      ISTOP = 0
162  DO 163 K = 1,LNCAT
      IF (N(K) - (PMIN + NOFEAT)) 167,163,163
163  CONTINUE
      IF (.NOT.DFL) RETURN
      DO 164 KK=1,LLNCAT
      DO 164 KKK=1,NOFEAT
164  MEANS(KKK,KK) = MEAN(KKK,KK)
      CALL PDDAT(MEANS,STDEV,N,CLD,IDAT1,IPLACE,AVP,AMN,MEANS)
      DO 165 KV=1,LLNCAT
      DO 165 KKK=1,NOFEAT
165  MEAN(KKK,KK) = MEANS(KKK,KK)
      CALL CLDIST(CLD,STDEV,MEANS)
      RETURN
167  WRITE(6,168) K,N(K),PMIN,NOFEAT
168  FORMAT(/' CLUSTER ',I3,' REMOVED FOR HAVING ONLY ',I6,' POINTS. '/
      *' MIN. POINTS IS (' ,I4,' * ',I2,' )')
      RETF=1
      LK=K
      GO TO 570
171  K=LK
      DEL = .TRUE.
      GO TO 162
169  CONTINUE
170  CONTINUE
C
C ON ITERATIONS 1 THRU N-1 CHECK N(K) AGAINST NMN
C
      DO 180 K=1,LNCAT
      IF (N(K)-NMN) 190,180,180
180  CONTINUE
      IF (DEL) CALL CLDIST(CLD,STDEV,MEANS)
      GO TO 220
190  IF (MOD(KKT,KRN)) 200,195,200
195  WRITE (6,210) K,N(K),NMN
200  RETF=2
      LK=K
      GO TO 570
201  K=LK
      DEL = .TRUE.
      GO TO 170
210  FORMAT(/' CLUSTER ',I2,' REMOVED FOR HAVING ONLY ',I6,
      *' ELEMENTS. MIN. NO. ELEMENTS IS ',I6)
220  CONTINUE
C
C
C* SPLIT ITERATION

```

```

5000800
5000810
5000820
5000830
5000840
5000850
5000860
5000870
5000880
5000890
5000900
5000910
5000920
5000930
5000940
5000950
5000960
5000970
5000980
5000990
5001000
5001010
5001020
5001030
5001040
5001050
5001060
5001070
5001080
5001090
5001100
5001110
5001120
5001130
5001140
5001150
5001160
5001170
5001180
5001190
5001200
5001210
5001220
5001230
5001240
5001250
5001260
5001270
5001280
5001290
5001300
5001310
5001320
5001330
5001340
5001350
5001360
5001370
5001380
5001390
5001400
5001410
5001420
5001430
5001440
5001450
5001460
5001470
5001480
5001490
5001500
5001510
5001520
5001530
5001540
5001550
5001560
5001570
5001580

```


FILE: ISOPAT

```

C
C
225 DO 225 I=1,INCAT
    PTR(I)=1
    ISPLT=0
    DO 260 K=1,INCAT
C*
C* FIND MAXIMUM STANDARD DEVIATION PER CLUSTER
C*
        SGMA(K) = 0.
        DO 250 J = 1,NOFEAT
            SDUM = STDEV(J,K)*SUNCOR(J)
            IF (SDUM - SGMA(K)) 250,240,240
240     SGMA(K) = J
            SGMA(K) = SDUM
250 CONTINUE
        IF (SGMA(K).GE.STUMAX) ISPLT=ISPLT+1
260 CONTINUE
        IF (2*INCAT.GT.MAXCL) CALDESCEN(SGMA,LNCAT,ISGMA,PTR)
        TEST=FLOAT(ISPLT)/FLOAT(LNCAT)
        IF (TEST.LE.PFWCEN) SPLFIN=1
        IF (KKT.GT.ISTOP) SPLFIN = 1
        IF (SPLFIN.EQ.0) GO TO 270
        IF (MOD(KKT,KRN).EQ.0) WRITE(6,503)
503 FORMAT(/)
        IF (MOD(KKT,KRN).EQ.0) WRITE(6,502)
502 FORMAT(1X,'USER INPUT*SPLIT-COMBINE SEQUENCE OF ITERATIONS')
        ISEQ=ISEQ+1
        IF (SEQUEN(ISEQ).EQ.SS) GO TO 270
        IF (SEQUEN(ISEQ).EQ.CC) GO TO 410
C
C IS SPLITTING REQUIRED
C
270 K=1
    NCAT=INCAT
    IF (K-NCAT) 290,290,500
290 IF (STUMAX-SGMA(K)) 300,300,310
300 IF (N(K)-(NMIN+NMN+2)) 310,310,320
310 K=K+1
    GO TO 280
C
C SPLIT CLUSTER K
C
320 TRIG1=1
    DEL=.TRUE.
    KX=ISGMA(K)
330 INCAT=INCAT+1
    LLNCAT=LLNCAT+1
    IF (LLNCAT.LE.MAXCLS) GO TO 350
    IF (MOD(KKT,KRN).EQ.0) WRITE(6,340) KKT
340 FORMAT(/,'MAXIMUM CLUSTERS ON ITERATION',I4,' SPLITTING REQUIRED R
    'UT NOT PERFORMED.'/)
    LNCAT = MAXCL
    LLNCAT=MAXCLS
    GO TO 500
350 INC=INCAT
    LL=PTR(K)
360 DO 370 I=1,KDIM
370 AMN(I,INC)=AMN(I,LL)
380 AMN(KX,LL)=AMN(KX,LL)+SEP*SGMA(K)
    AMN(KX,INC)=AMN(KX,INC)-SEP*SGMA(K)
    SGMA(K)=0.0
    IF (MOD(KKT,KRN)) 400,401,400
401 WRITE(6,390) LL,KX,INC
390 FORMAT('O CLUSTER ',I2,' IS SPLIT IN THE ',I2,'TH PARAMETER INTO C
    LUSTER ',I2)
400 CONTINUE
    K=K+1
    GO TO 280
C
C EVEN ITERATION
C
C ARE CLUSTERS TO BE COMBINED
C
410 CONTINUE
    DO 405 L=1,LNCAT
405 PTR(L)=1
C
    NOCOMB=0

```

FILE: ISOPAT

```

      NOCLST=LNCAT-1
      L=1
406  L=L+2
      IF (L.GT.NOCLST) GO TO 480
      NOCLTR = LNCAT - 1
      KK=0
      DMIN=DMIN
      DO 430 I=1,NOCLTR
C      IF (PTR(I).EQ.0) GO TO 430
      II=1
      DO 425 J=II,LNCAT
      IF (PTR(J).EQ.0) GO TO 425
      SDIJ = 0.0
      DO 420 JJ=1,KDIM
      SDIJ=SDIJ+((AMN(JJ,I)-AMN(JJ,J))*2/(STDEV(JJ,I)*STDEV(JJ,J)))
420  CONTINUE
      DIJ=SDIJ
C      IF (DIJ.GT.DMIN) GO TO 425
      DMIN=DIJ
      KK=I
      KKK=J
425  CONTINUE
430  CONTINUE
C      IF (KK.EQ.0) GO TO 480
      PTR(KK)=0
C      COMBINE CLUSTERS KK AND KKK
      DEL=TRUE
      RND=1.0 /FLOAT(N(KK)+N(KKK))
C      DO 460 K=1,KDIM
460  AMN(K,KK)=(N(KK)*AMN(K,KK)+N(KKK)*AMN(K,KKK))*RND
C      RETF=3
      LK=KKK
      GO TO 570
461  KKK=LK
      IF (KKK.EQ.(LNCAT+1)) GO TO 435
C      MOVE POINTERS UP
      DO 175 K=KKK,LNCAT
      PTR(K) = PTR(K+1)
C      175
435  IF (MOD(KKT,KRN))440,441,440
441  WRITE(A.490)KK,KKK,KK
440  IF (L.LT.NOCLST) GO TO 406
C      CONTINUE
480  FORMAT(' CLUSTERS '.I2.' AND '.I2.' HAVE BEEN COMBINED INTO CLUST
      2ER '.I2)
C      REINITIALIZE
C      500 CONTINUE
      DO 510 J=1,MAXCLS
      SGMA(J)=0.0
      ISGMA(J)=0
      DO 510 K=1,KDIM
      AVP(K,J)=0.0
      STDEV(K,J)=0.0
      MEANS(K,J)=AMN(K,J)
      AMN(K,J)=0.0
510  CONTINUE
      KKT=KKT+1
      DEL=.FALSE.
      GO TO 10
C      530 IF (KKT.NE.2) GO TO 550
      WRITE (A.540)
540  FORMAT(' THE ORIGINAL CLUSTER WAS NOT SPLIT - EXAMINE THE INPUT VA
      LUE FOR STOMAXI/')
      KKT=1
      ISTOP=0
      GO TO 10

```

FILE: ISOPAT

```

550 WRITE (6,560)KKT
560 FORMAT(//: AFTER ',14.' ITERATIONS ALL DATA HAS BEEN ASSIGNED TO 0
      *NF CLUSTER'/)
      KKT=1
      ISTOP=0
      GO TO 10
570 CONTINUE
C*
C* ROUTINE TO DELETE A CLUSTER
C*
      INCAT=INCAT-1
      LLNCAT=LLNCAT-1
      IF (LK.EQ.(INCAT-1).AND.DODU.EQ.0) GO TO (171,201,461),RETF
      DO 561 J=LK,LLNCAT
      DO 552 L=1,KHIM
      AMN(L,J)=AMN(L,J-1)
      MEANS(L,J)=MEANS(L,J-1)
      MEAN(L,J)=MEAN(L,J-1)
552 STDEV(L,J)=STDEV(L,J-1)
      N(J)=N(J-1)
561 CONTINUE
      GO TO (171,201,461),RETF
      END

```

5003170
5003180
5003190
5003200
5003210
5003220
5003230
5003240
5003250
5003260
5003270
5003280
5003290
5003300
5003310
5003320
5003330
5003340
5003350
5003360
5003370
5003380
5003390

၁၁၁

C

5
C
C
C

20

25

30

40

41

PSP00010
PSP00020
PSP00030
PSP00040
PSP00050
PSP00060
PSP00070
PSP00080
PSP00090
PSP00100
PSP00110
PSP00120
PSP00130
PSP00140
PSP00150
PSP00160
PSP00170
PSP00180
PSP00190
PSP00200
PSP00210
PSP00220
PSP00230
PSP00240
PSP00250
PSP00260
PSP00270
PSP00280
PSP00290
PSP00300
PSP00310
PSP00320
PSP00330
PSP00340
PSP00350
PSP00360
PSP00370
PSP00380
PSP00390
PSP00400
PSP00410
PSP00420
PSP00430
PSP00440
PSP00450
PSP00460
PSP00470
PSP00480
PSP00490
PSP00500
PSP00510
PSP00520
PSP00530
PSP00540
PSP00550
PSP00560
PSP00570
PSP00580
PSP00590
PSP00600
PSP00610
PSP00620
PSP00630
PSP00640
PSP00650
PSP00660
PSP00670
PSP00680
PSP00690
PSP00700
PSP00710
PSP00720
PSP00730
PSP00740
PSP00750
PSP00760
PSP00770
PSP00780
PSP00790

FILE: PSPPAT

```

42  KK=1
    SDIST=10.0E+20
    DO 46 J=1,LNCAT
    DIST=0.
    DO 44 K=1,NOFEAT
        LPACK(4) = LARRAY(IBASE + K)
        CSUN(K) = IPACK(1)
44  DIST=DIST+ARS(MEANS(K,J)-CSUN(K))*SUNCOR(K)
    IF (DIST - SDIST) 45,46,46
45  KK=J
    SDIST=DIST
    CONTINUE
46  CONTINUE
47  CONTINUE
    N(KK)=N(KK)+1
    IPLACE(1)=KK
    DO 48 K=1,NOFEAT
        AMN(K,KK)=AMN(K,KK)+CSUN(K)
        AVP(K,KK)=AVP(K,KK)+CSUN(K)**2
48  CONTINUE
49  CONTINUE
    GO TO 101
50  DO 100 I = 1,ICCT
        IBASE = (I - 1)*NOFEAT + 1B
        KK=1
        IF (DODU.EQ.0) GO TO 52
        DO 51 K=1,NOFEAT
            LPACK(4) = LARRAY(IBASE + K)
            CDUM = IPACK(1)
            IF (CDUM.NE.MEANDO.AND.CDUM.NE.MEANDU) GO TO 52
51  CONTINUE
            IF (CDUM.EQ.MEANDO) IPLACE(1) = LNCAT + 1
            IF (CDUM.EQ.MEANDU) IPLACE(1) = LNCAT + DODU
        GO TO 100
52  CONTINUE
        KK = 1
        SDIST=10.0E+20
        DO 70 J=1,LNCAT
        DIST=0.0
        DO 55 K = 1,NOFEAT
            LPACK(4) = LARRAY(IBASE + K)
            CSUN(K) = IPACK(1)
55  DIST = DIST + ARS(MEANS(K,J) - CSUN(K))
        IF (DIST-SDIST) 60,70,70
60  KK=J
        SDIST=DIST
70  CONTINUE
80  CONTINUE
        N(KK)=N(KK)+1
        IPLACE(1)=KK
        DO 90 K=1,NOFEAT
            AMN(K,KK) = AMN(K,KK) + CSUN(K)
            AVP(K,KK) = AVP(K,KK) + CSUN(K)**2
90  CONTINUE
100 CONTINUE
101 CONTINUE
        IF (IRC.EQ.0) GO TO 110
        CALL RWRITE (ADRES2,IPLACE,ICCT,ISTAT)
        ADRES2=ADRES2+ICCT
105 IF (ISTAT.EQ.1) GO TO 105
110 IRC=IRC-1
        IF (IRC.GT.0) GO TO 20
        KA = 1
115 CONTINUE
        DO 130 K=KA,LNCAT
        IF (N(K).EQ.0) GO TO 130
        RND=FLOAT(N(K))
        DO 130 J=1,NOFEAT
            AMN(J,K)=AMN(J,K)/RND
            MEANS(J,K)=AMN(J,K)
            STDEV(J,K)=SORT(AVP(J,K)/RND-AMN(J,K)*AMN(J,K))
            DUMA = STDEV(J,K)
            IF (DUMA.LT.DUM) STDEV(J,K) = DUM
130 CONTINUE
        RETURN
    END

```

PSP00800
PSP00810
PSP00820
PSP00830
PSP00840
PSP00850
PSP00860
PSP00870
PSP00880
PSP00890
PSP00900
PSP00910
PSP00920
PSP00930
PSP00940
PSP00950
PSP00960
PSP00970
PSP00980
PSP00990
PSP01000
PSP01010
PSP01020
PSP01030
PSP01040
PSP01050
PSP01060
PSP01070
PSP01080
PSP01090
PSP01100
PSP01110
PSP01120
PSP01130
PSP01140
PSP01150
PSP01160
PSP01170
PSP01180
PSP01190
PSP01200
PSP01210
PSP01220
PSP01230
PSP01240
PSP01250
PSP01260
PSP01270
PSP01280
PSP01290
PSP01300
PSP01310
PSP01320
PSP01330
PSP01340
PSP01350
PSP01360
PSP01370
PSP01380
PSP01390
PSP01400
PSP01410
PSP01420
PSP01430
PSP01440
PSP01450
PSP01460
PSP01470
PSP01480
PSP01490
PSP01500
PSP01510
PSP01520
PSP01530
PSP01540

ORIGINAL PAGE 10
OF POOR QUALITY

FILE: RDDPAT

```

C* THIS SUBROUTINE COORDINATES THE ROUTINES TO READ FIELDS OF DATA FROM THE IMAGE TAPE AND STORE IT ON A DRUM FILE FOR THE ISOCLC ROUTINES.
C*
C* SUBROUTINE RDDPAT(FD1,TOP,IDATA,IDIM,LAST)
C* IMPLICIT INTEGER (A-Z)
C* DIMENSION FLDINF(6),IDATA(10IM),FL(12),LSTAT(3)
C* COMMON ARRAY(10600)
C* LOGICAL*1 LARRAY(42400)
C* EQUIVALENCE (ARRAY,LARRAY)
C* LOGICAL*1 LPACK(4)
C* EQUIVALENCE (LPACK,IPACK)
C* INCLUDE COMMONS.LIST
C* INCLUDE COMMONS.LIST
C* COMMON/PASS/STOP,LNCAT,NMIN,KRY,STOMAX,DLMIN,SEP,
C* MAP,SPTPIG, IFO, KPTS, NOPTS, PUNCH,
C* ICHN,CHNTHS,ICHAIN(62),NPOS,IREGIN,REGIN1,
C* REGIN2,REGIN3,CLSLAM,NOFLD,IPT,TOTWRD,TOTPTS,
C* NCLASS,NOCLS,TOTISU,TOTFLD,TOTVRT,NOCL,NVRT
C* ,NXTCLS,NOFFAT,MAXCLS,FFTVFC(30),SYMATX(62)
C* ,VARSI2,STATKY,TSKEY,4APFMT,MAPKEY,SEQUEN(20),PERCEN,SIMERP
C* ,IORDER,INOUT,INFILE,INIT4,PRIN,SUSVEC(62),NOSUB2,CHNVC(30)
C* ,NOCHAN,FRCOMP,NOSE0,REAND0,REANDU,
C* SYND0,SYNDU,ITRIG0,ITRIGU,DOFLAG,
C* DOFLAG,NOU,STOTS(60),NSPTS,SINCOR(30),LLNCAT,
C* DVERT(250,2),DRECT(60,2),DVPLT(11,2),IDCN1(2),NDOU(2)
C* ,MAXFT1,MAXPOP
C* REAL SINCOR
C* COMMON/GLOBAL/HEAD(63),MARTAP,DATAPE,SAVTAP,BMFILE,HMKEY,
C* ,HISFIL,HISKEY,IPFORM,EWIPT,EPKEY,MAPUNT,NOFILE,
C* ,OPIMAN,OPMANS,PASSIZ,DATEFIL,STAFIL,ASAV,ASAVFL
C* ,NHSTUN,NHSTFL,SCRUN,MAPFIL
C* ,DOTUNT,DOTFIL,NCHPAS,TRNSFL,RMTREL,HISTFL,PCHUNT,
C* CRDUNT,PRUNT,RANDIO
CSEND
  PRUNIT = 30
  DIMENSION C490(20)
  EQUIVALENCE (FLDINF(1),LINSTR), (FLDINF(4),SAMSTR),
  (FLDINF(2),LINEAD), (FLDINF(5),SAMEND),
  (FLDINF(3),LININC), (FLDINF(6),SAMINC)
  DATA LORN/1/
  DIMENSION LND(2),LDOU(11,2),ID3(35),IDE(35),NDINT(11,2),
  DINT(120,2),DPIST(12,2),DIN(70)
  DATA DONAME/10THE/
  DATA DUNAME/10TID/
  DIMENSION FLDSAV(4,10),VERTEX(220)

C* RESERVE 2000 LOCATIONS OF 'ARRAY' FOR FIELD DEFINITION INFORMATION.
C* THE REMAINDER OF 'ARRAY' IS USED FOR I/O BUFFERS.
C*
C* CLASS AND FIELD INFORMATION STORED AS FOLLOWS
C*
C* ARRAY(1) =CLASS NAME
C* ARRAY(2) =RESERVED FOR INDEX POINTER TO NEXT CLASS NAME
C* ARRAY(3) =RESERVED FOR NO. OF CLUSTERS IN THIS CLASS
C* ARRAY(4) =NO. OF FIELDS FOR THIS CLASS
C* ARRAY(5) =FIRST FIELD NAME FOR THIS CLASS
C* (6) =NO. OF VERTICES FOR THIS FIELD (NV)
C* (7)-(7+NV*2) = ACTUAL VERTEX NUMBERS
C* (8+NV*2) =TOTAL PIXELS IN THIS FIELD
C* (9+NV*2)-(10+NV*2) = FLDINF BLOCK FOR THIS FIELD
C*
C* CALL TA=HDD(DATAPE,DATEFIL)
C* CONTINUE
C* RESECV=2000
C* ADDRES=REGIN1
C* ICHN=1
C* INPT=0
C* NVRT=0
C* LAST=0
C* TOTWRD=0
C* IDP = 0
C* DOFLAG = 0
C* DOFLAG = 0
C* DDOU = 0
C* NDOU(1) = 0
C* NDOU(2) = 0
C* 2 REINDEX=RESECV+1
C* NHUFS=1

```

RDD00010
 RDD00020
 RDD00030
 RDD00040
 RDD00050
 RDD00060
 RDD00070
 RDD00080
 RDD00090
 RDD00100
 RDD00110
 RDD00120
 RDD00130
 RDD00140
 RDD00150
 RDD00160
 RDD00170
 RDD00180
 RDD00190
 RDD00200
 RDD00210
 RDD00220
 RDD00230
 RDD00240
 RDD00250
 RDD00260
 RDD00270
 RDD00280
 RDD00290
 RDD00300
 RDD00310
 RDD00320
 RDD00330
 RDD00340
 RDD00350
 RDD00360
 RDD00370
 RDD00380
 RDD00390
 RDD00400
 RDD00410
 RDD00420
 RDD00430
 RDD00440
 RDD00450
 RDD00460
 RDD00470
 RDD00480
 RDD00490
 RDD00500
 RDD00510
 RDD00520
 RDD00530
 RDD00540
 RDD00550
 RDD00560
 RDD00570
 RDD00580
 RDD00590
 RDD00600
 RDD00610
 RDD00620
 RDD00630
 RDD00640
 RDD00650
 RDD00660
 RDD00670
 RDD00680
 RDD00690
 RDD00700
 RDD00710
 RDD00720
 RDD00730
 RDD00740
 RDD00750
 RDD00760
 RDD00770
 RDD00780
 RDD00790

FILE: RDDPAT

```

MAXDIM=TOP-RESERV
HUESI7= MAXDIM/(NHUES*NOFEAT) * NOFEAT
HUESI7 = (HUESI7/4)*4
IF (HUESI7.GT. 100) GO TO 3
RESERV=RESERV-100
IF (RESERV.GT.30) GO TO 2
GO TO 70
3 CONTINUE
NOFL(0)=0
IPT=1 + F01 - 1
TOTVT2=0
IF (NOCL.F0.0) GO TO 5
4 ARRAY(IPT)=NXTCLS
IPT=IPT+4
WRITE(6,HEAD)
WRITE(6,500)NXTCLS
C*
C* READ A FIELD DESCRIPTION FROM CARDS.
C*
5 ICK = LAREAD(ARRAY(IPT),ARRAY(IPT+2),FLDINF,ARRAY(IPT+1) )
IF (ICK.NE.-3) GO TO 1000
WRITE(6,140)
READ (RUNIT,150) (CARD(I), I=1,20)
WRITE(6,150) (CARD(I), I=1,20)
150 FORMAT(20A4)
160 FORMAT(1X,20A4)
REWIND RUNIT
IDP=IDP+1
IDCNT(IDP)=0
OVPNT(1,IDP)=1
READ(30,100) DNAME
REWIND 30
IF (DNAME.F0.DNAME) ITRIG0=1
IF (DNAME.F0.DNAME) ITRIGU=1
IF (DNAME.F0.DNAME) IS=2
IF (DNAME.F0.DNAME) IS=1
INDV=1
INDP=1
GO TO 5
1000 IF (ICK.LE.0.OR.IDP.LE.0) GO TO 1030
IF (IDCNT(IDP).LT.10) GO TO 1025
WRITE(6,170)
170 FORMAT(// * TOO MANY DO OR DU FIELDS THESE IGNORED*)
GO TO 5
1025 CONTINUE
READ (RUNIT,150) (CARD(I), I=1,20)
WRITE(6,160) (CARD(I), I=1,20)
REWIND RUNIT
OVERT(INDV,IDP) = ARRAY(IPT + 1)
IDLIM = OVERT(INDV,IDP)*2
DO 1010 I=1,IDLIM
INDV=INDV+1
VERTX(TOTVT2+1)=ARRAY(IPT+1+1)
OVERT(INDV,IDP) = ARRAY(IPT + I + 1)
1010 INDV = INDV + 1
TOTVT2=TOTVT2+IDLIM
DO 1020 I=1,6
OVERT(INDP,IDP)=FLDINF(I)
1020 INDP=INDP+1
IDCNT(IDP)=IDCNT(IDP)+1
IDUP=IDCNT(IDP)+1
OVPNT(IDUP,IDP)=INDV
GO TO 5
C 1030 FINISHED WITH DO/OU FIELD PROCESSING
CONTINUE
IDP=0
IDUP=IT-IGO+ITRIGU
IF (ICK.LT.0) GO TO 20
IF (ICK.F0.0) GO TO 30
IF (NOCL.GT.0) GO TO 6
WRITE(6,400)
CALL CHRP
6 CONTINUE
NV=ARRAY(IPT+1)
NVIT=NVIT+NV
NOFL=NOFL+1
NSAMP=(SAMEND-SAMSTR)/SAMINC+1
FLOSAM=0
I=IPT+2

```

RDD00800
 RDD00810
 RDD00820
 RDD00830
 RDD00840
 RDD00850
 RDD00860
 RDD00870
 RDD00880
 RDD00890
 RDD00900
 RDD00910
 RDD00920
 RDD00930
 RDD00940
 RDD00950
 RDD00960
 RDD00970
 RDD00980
 RDD00990
 RDD01000
 RDD01010
 RDD01020
 RDD01030
 RDD01040
 RDD01050
 RDD01060
 RDD01070
 RDD01080
 RDD01090
 RDD01100
 RDD01110
 RDD01120
 RDD01130
 RDD01140
 RDD01150
 RDD01160
 RDD01170
 RDD01180
 RDD01190
 RDD01200
 RDD01210
 RDD01220
 RDD01230
 RDD01240
 RDD01250
 RDD01260
 RDD01270
 RDD01280
 RDD01290
 RDD01300
 RDD01310
 RDD01320
 RDD01330
 RDD01340
 RDD01350
 RDD01360
 RDD01370
 RDD01380
 RDD01390
 RDD01400
 RDD01410
 RDD01420
 RDD01430
 RDD01440
 RDD01450
 RDD01460
 RDD01470
 RDD01480
 RDD01490
 RDD01500
 RDD01510
 RDD01520
 RDD01530
 RDD01540
 RDD01550
 RDD01560
 RDD01570
 RDD01580

FILE: RDDPAT

```

      NQ=NV-1
      NQ=NO-5
      IF (NQ.GT.5) NQ=5
      IF=IH+NQ*2 - 1
      WRITE (A,600) NOFLD,ARRAY(IPT),SAMINC,LININC,
      * (LPRN,ARRAY(I),ARRAY(I+1),I=IH,IE,2)
      IF (NR.LE.0) GO TO 7
      IR=IF+1
      IF=IH+NQ*2 - 1
      WRITE (A,650) (LPRN,ARRAY(I),ARRAY(I+1),I=IB,IE,2)
7 CONTINUE
      IF (NSAMP*NOFEAT.GT.IDIM) GO TO 90
C* POSITION TAPE FOR THIS FIELD
C* CALL FLDINT(FLDINF,FETVEC,NOFEAT)
      FLDOSAM=0
      DO 10 LINE=LINSTR,LINEND,LININC
      LND(1)=0
      LND(2)=0
      IDRR = 2
      IDEF=1
      IF (IDRR.EQ.0) GO TO 1095
C BOTH DO AND DU TRIGGERS OFF --- SKIP AROUND
      DO 1060 IDP=1,IDRR
      IDLIM=IDRR*(IDP)
      DO 1050 I=1,IDLIM
      IDUM=(I-1)*6
      LDSTR=OVERT(IDUM+1,IND)
      LDEND=OVERT(IDUM+2,IND)
      LDINC=OVERT(IDUM+3,IND)
      DO 1040 IT = LDSTR,LDEND,LDINC
      IF (IT.NE.I) IF GO TO 1040
      LND(IND)=LND(IND)+1
      IDUM=LND(IND)
      LDOU(IDUM,IND) = I
1040 CONTINUE
1050 CONTINUE
1060 CONTINUE
      IF (LND(1).EQ.0.AND.LND(IDRR).EQ.0) GO TO 1095
C NO DO OR DU FOR THIS LINE
      IF (LND(1).GT.0) IDRR=1
      IF (IDRR.EQ.2.AND.LND(2).GT.0) IDEF=2
      DO 1090 IDP=IDRR,IDEF
      IDLIM=LND(IDP)
      IF (IDLIM.EQ.0) GO TO 1090
      IDUM=0
      DOINT(1,1) = 1
      DOINT(1,2) = 1
      DO 1080 I=1,IDLIM
      IDP=DOINT(1,IND)
      OVP=OVERT(IDP,IND)
      CALL FLDINT(OVERT(OVP+1,IND),OVERT(OVP,IND),FL,LINE,SAMPS,NI)
      DOINT(1,IND)=NI
      IF (NI.EQ.0) GO TO 1080
      DO 1070 II=1,NI
      DOINT(II+1,OVP,IND)=FL(II)
      IDUM=IDUM+NI
      DOINT(II+1,IND) = IDUM + 1
1070 CONTINUE
1080 CONTINUE
1090 CONTINUE
1095 CALL LINEED(IDATA,ENDTAP)
      IF (ENDTAP.EQ.-1) GO TO 80
C* FIND SAMPLE INTERSECTS FOR THIS LINE - NI=NO. OF INTERSECTS
C* CALL FLDINT(ARRAY(IPT+2),IV,FL,LINE,SAMPS,NI)
C* STORE DATA ON THIS LINE INTO OUTPUT BUFFER
C* MODSS=MOD(SAMSTR,SAMINC)
      DO 60 I=1,NI*2
      IP=(FL(I)-SAMSTR)/SAMINC+1
      IF=(FL(I+1)-SAMSTR)/SAMINC+1
      IF (MODSS.NE.MOD(FL(I),SAMINC)) IP=IP+1
      IF (IP.GT.IE) GO TO 60
      IF (IDRR.EQ.0) GO TO 2055
      IF (LND(IDRR).EQ.0.AND.LND(IDEF).EQ.0) GO TO 2055
      RDD01590
      RDD01600
      RDD01610
      RDD01620
      RDD01630
      RDD01640
      RDD01650
      RDD01660
      RDD01670
      RDD01680
      RDD01690
      RDD01700
      RDD01710
      RDD01720
      RDD01730
      RDD01740
      RDD01750
      RDD01760
      RDD01770
      RDD01780
      RDD01790
      RDD01800
      RDD01810
      RDD01820
      RDD01830
      RDD01840
      RDD01850
      RDD01860
      RDD01870
      RDD01880
      RDD01890
      RDD01900
      RDD01910
      RDD01920
      RDD01930
      RDD01940
      RDD01950
      RDD01960
      RDD01970
      RDD01980
      RDD01990
      RDD02000
      RDD02010
      RDD02020
      RDD02030
      RDD02040
      RDD02050
      RDD02060
      RDD02070
      RDD02080
      RDD02090
      RDD02100
      RDD02110
      RDD02120
      RDD02130
      RDD02140
      RDD02150
      RDD02160
      RDD02170
      RDD02180
      RDD02190
      RDD02200
      RDD02210
      RDD02220
      RDD02230
      RDD02240
      RDD02250
      RDD02260
      RDD02270
      RDD02280
      RDD02290
      RDD02300
      RDD02310
      RDD02320
      RDD02330
      RDD02340
      RDD02350
      RDD02360
      RDD02370

```


FILE: RDDPAT

ORIGINAL PAGE IS
OF POOR QUALITY

```

DO 2050 I=0,IOBR,IOEE
  IDLIM=IND(I)
  IF (IDLIM.EQ.0) GOTO 2050
  IDSIT=1
  MEANDD=MEANDD
  IF (IDPR.EQ.IDFE) GOTO 2003
  IF (IND.EQ.2) IDSIT=2
  IF (IND.EQ.2) MEANDD=MEANDD
  GOTO 2009
2003 IF (IDPR.EQ.1.AND.IDPP.EQ.2) GOTO 2009
  IF (ITPRIGU.EQ.0) GOTO 2009
  IDSIT=2
  MEANDD=MEANDD
2009 CONTINUE
  DO 2040 K=1,IDLIM
    MDIN=NDINT(K,IND)
    IF (MDIN.EQ.0) GOTO 2040
    DPIN=DPINT(K,IND)
    DO 2010 KK=1,MDIN
      GTN(KK)=DINT(DPIN+KK-1,IND)
      IDUM=0
      DO 2020 KX=1,NDIN,2
        IDUM=IDUM+1
        IDR(IDUM)=(DIN(KK)-SAMSTR)/SAMINC+1
        IDF(IDUM)=(DIN(KK+1)-SAMSTR)/SAMINC+1
        IF (MODSS.NE.MOD(DIN(KK),SAMINC)) ID3(IDUM)=IDR(IDUM)+1
      CONTINUE
      DO 2030 KK=1,IDUM
        IDS=IDR(KK)
        IDF=IDF(KK)
        IF (IDS.GT.IDR.IDR.IDR.GT.IDF) GOTO 2030
        IF (IDS.LE.IDR) IDS=IDR
        IF (IDF.GT.IDR) IDF=IDR
        IF (IDS.GT.IDF) GOTO 2030
        DO 2025 KKK=IDS,IDF
          ND0U(IND)=ND0U(IND)+1
          DO 2023 KKKK=1,NOFEAT
            DUMMY1=KKK+NSAMP*(KKKK-1)
            IDATA(DUMMY1)=MEANDD
          CONTINUE
        CONTINUE
        CONTINUE
        CONTINUE
        IF (IDSIT.EQ.1.AND.ND0U(IND).GT.0) DOFLAG=1
        IF (IDSIT.EQ.2.AND.ND0U(IND).GT.0) DUFLAG=1
      CONTINUE
      DO 2055 DOFLAG=DOFLAG+DUFLAG
        IRASF = (IRINDEX - 1)*4
        IRUF4 = IRFSIZ*4
        IRWS4 = IRWS*4
        DO 50 J=1,IF
          FLD0SAM=FLDSAM+1
          DO 50 K=1,NOFEAT
            IWRD=IWRD+1
            DUMMY2=J+NSAMP*(K-1)
            IPACK = IDATA(DUMMY2)
            DUMMY3 = IRASF + IWRD
            LARRAY(DUMMY3) = IPACK(4)
            IF (IWRD.LT.IRUF4) GO TO 50
            TOTWRD = TOTWRD + IWRD
            IF (TOTWRD.GT.IRWS4) GO TO 35
            CALL WRITE(ADDRESS,ARRAY(IRINDEX),IRFSIZ,LSTAT(1))
            ADDRESS = ADDRESS + IRFSIZ
          CONTINUE
        IF (LSTAT(IRUF).EQ.1) GOTO 40
        IWRD=0
      CONTINUE
    CONTINUE
  CONTINUE
  IPT = IPT + NV*2 + 2
  AWRAY(IPT)=FLDSAM
  DO 15 I=1,4
    IPT=IPT+1
  AWRAY(IPT)=FLDINF(I)
  IPT=IPT+1
  IF (IPT+30.GT.RESERV) GO TO 70
  GO TO 5

```

C* CLASS NAME CARD ENCOUNTERED - REFEAD PREVIOUS CARD TO GET NAME
C*
20 NOCL=NOCL+1

RDD02380
 RDD02390
 RDD02400
 RDD02410
 RDD02420
 RDD02430
 RDD02440
 RDD02450
 RDD02460
 RDD02470
 RDD02480
 RDD02490
 RDD02500
 RDD02510
 RDD02520
 RDD02530
 RDD02540
 RDD02550
 RDD02560
 RDD02570
 RDD02580
 RDD02590
 RDD02600
 RDD02610
 RDD02620
 RDD02630
 RDD02640
 RDD02650
 RDD02660
 RDD02670
 RDD02680
 RDD02690
 RDD02700
 RDD02710
 RDD02720
 RDD02730
 RDD02740
 RDD02750
 RDD02760
 RDD02770
 RDD02780
 RDD02790
 RDD02800
 RDD02810
 RDD02820
 RDD02830
 RDD02840
 RDD02850
 RDD02860
 RDD02870
 RDD02880
 RDD02890
 RDD02900
 RDD02910
 RDD02920
 RDD02930
 RDD02940
 RDD02950
 RDD02960
 RDD02970
 RDD02980
 RDD02990
 RDD03000
 RDD03010
 RDD03020
 RDD03030
 RDD03040
 RDD03050
 RDD03060
 RDD03070
 RDD03080
 RDD03090
 RDD03100
 RDD03110
 RDD03120
 RDD03130
 RDD03140
 RDD03150
 RDD03160

FILE: RDPAT

| | |
|---|----------|
| IF (NOCL.GT.1) GO TO 25 | RDD03170 |
| READ(30,100)NXTCLS | RDD03180 |
| REWIND 30 | RDD03190 |
| GO TO 4 | RDD03200 |
| 25 CLSNAM=NXTCLS | RDD03210 |
| READ(30,100)NXTCLS | RDD03220 |
| REWIND 30 | RDD03230 |
| GO TO 31 | RDD03240 |
| C* EMPTY LAST BUFFER AND RETURN TO PROCESS DATA FOR THIS CLASS. | RDD03250 |
| 30 CLSNAM=WXCLS | RDD03260 |
| LAST=1 | RDD03270 |
| C* | RDD03280 |
| 31 TOTWRD=TOTWRD+IWRD | RDD03290 |
| IF (TOTWRD.GT.NWDS) GO TO 35 | RDD03300 |
| IWRD4 = IWRD/4 | RDD03310 |
| IF (4*IWRD4.NE.IWRD) IWRD4 = IWRD4 + 1 | RDD03320 |
| CALL WRITE(AADRES,ARRAY(4*FINDA),IWRD4,LSTAT(1)) | RDD03330 |
| TOTPTS=TOTWRD/NOFEAT | RDD03340 |
| IPT = IPT - FDI + 1 | RDD03350 |
| IPAT = (TOTWRD/4) + 1 | RDD03360 |
| IF (IPAT.TOTPTS.LE.NWDS) RETURN | RDD03370 |
| WRITE(6,200)NCLS | RDD03380 |
| IPT = IPT - FDI + 1 | RDD03390 |
| RETURN | RDD03400 |
| 35 WRITE(6,200)NWDS | RDD03410 |
| CALL CMFRR | RDD03420 |
| 70 WRITE(6,300)RESERV | RDD03430 |
| CALL CMFRR | RDD03440 |
| 90 WRITE(6,400) | RDD03450 |
| CALL CMFRR | RDD03460 |
| 90 WRITE(6,700)IDIM | RDD03470 |
| CALL CMFRR | RDD03480 |
| 100 FORMAT(10X,A4) | RDD03490 |
| 140 FORMAT(/// * DESIGNATED OTHER OR UNIDENTIFIABLE FIELDS INPUT'//) | RDD03500 |
| 200 FORMAT(' * TOO MUCH DATA REQUESTED--PIXELS*(CHANNELS+1) CANNOT EXCE' | RDD03510 |
| *0',I10) | RDD03520 |
| 300 FORMAT(' * STORAGE REQUIRED FOR FIELD DEFINITION INFORMATION EXCEEDS' | RDD03530 |
| * THE DIMENSION LIMIT OF',I5) | RDD03540 |
| 400 FORMAT(' * END-OF-TAPE REACHED BEFORE END OF FIELD') | RDD03550 |
| 500 FORMAT(//40X,'FIELDS TO BE CLUSTERED FOR CLASS',1X,A4// | RDD03560 |
| * T35,'SAMPLE',T45,'LINE',/T20,'FIELD NAME',T36,'INC.', | RDD03570 |
| * T45,'INC.',T73,'VERTICES (SAMPLE,LINE)')// | RDD03580 |
| 600 FORMAT(1X,T16,T3,T22,T44,T36,T4,T45,T14,T60, | RDD03590 |
| * 5(31,T4,'',T4,'')',1X)) | RDD03600 |
| 650 FORMAT(1X,T60,5(A1,T4,'',T4,'')',1X)) | RDD03610 |
| 700 FORMAT(' * NO. OF PIXELS TO BE UNPACKED PER SCAN EXCEEDS THE DIMENS' | RDD03620 |
| *ON LIMIT OF',I5) | RDD03630 |
| 800 FORMAT(// * INPUT ERROR - A CLASSNAME CARD MUST BE INPUT BEFORE A G' | RDD03640 |
| *ROUP OF FIELDS'//) | RDD03650 |
| RETURN | RDD03660 |
| END | RDD03670 |